

# **RANCANG BANGUN APLIKASI PENGELOLAAN BAHAN BAKU KAFE MENGUNAKAN METODE *ANALYTICAL HIERARCHY PROCESS* (AHP) DAN *PUSH NOTIFICATION***

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Julita Gandasari Ariana

NIM: 115060801111061



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## PENGESAHAN

RANCANG BANGUN APLIKASI PENGELOLAAN BAHAN BAKU KAFE  
MENGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS (AHP) DAN PUSH  
NOTIFICATION

### SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Julita Gandasari Ariana  
NIM: 115060801111061

Skripsi ini telah diuji dan dinyatakan lulus pada  
3 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Aryo Pinandito, S.T, M.MT  
NIP. 198305192014041001

Muhammad Tanzil Furgon, S.Kom, M.CompSc  
NIP. 19820930 200801 1 004

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoro Kurniawan, S.T, M.T, Ph.D  
NIP. 19710518 200312 1 001

## IDENTITAS TIM PENGUJI

Penguji 1:

Agi Putra Kharisma, S.T, M.T

NIP/NIK: 201304 860430 1 001

Penguji 2:

Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.

NIP/NIK. 2016078912041001



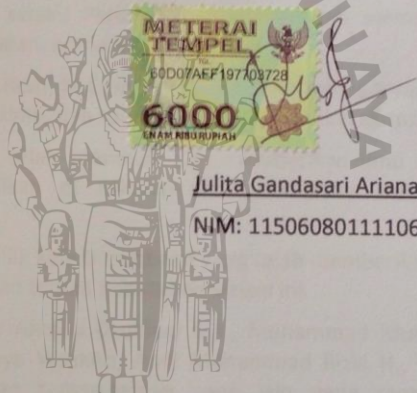


## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Agustus 2018



Julita Gandasari Ariana

NIM: 115060801111061

## DAFTAR RIWAYAT HIDUP

Nama : Julita Gandasari Ariana  
 Tempat, Tanggal Lahir : Malang, 5Juli 1993  
 Riwayat Sekolah : SD Negeri Percobaan Malang  
                               SMP Negeri 1 Malang  
                               SMA Negeri 1 Malang



## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT atas anugerah dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Aplikasi Pengelolaan Bahan Baku Kafe Menggunakan Metode *Analytical Hierarchy Process* (AHP) dan *Push Notification*” ini. Skripsi ini disusun sebagai syarat memperoleh gelar sarjana pada Program Studi Teknik Informatika, Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.

Dalam proses penyelesaian skripsi ini penulis mendapatkan banyak bantuan baik moral maupun materiil dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua, yaitu Drs. Arief Susanto Ak., CES dan Sri Harini beserta anggota keluarga yang lain yang telah memberikan dukungan dengan segala usaha dan doa selama ini.
2. Aryo Pinandito, S.T, M.MT selaku Pembimbing I yang telah memberikan bimbingan, ilmu, dan saran dalam penyelesaian skripsi ini.
3. Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku Pembimbing II yang juga telah memerikan bimbingan, ilmu, dan saran dalam penyelesaian skripsi ini.
4. Bapak dan Ibu Dosen yang telah mendidik dan memberikan ilmu selama penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
5. Ferdha Agisyanto selaku pemilik kafe Bunchbead yang telah memberikan data dan penjelasan yang diperlukan terkait kebutuhan skripsi ini.
6. Teman-teman terdekat yaitu Ahmad Dzulfikar S.R., Muhammad Murtadho, Rizky Kharisma, Stefanus Bayu Waskito, Andi Mohammad Rizki H., I Gusti Ngurah Ersania Susena, dan teman-teman yang lain yang membantu memberikan saran dalam pengerjaan dan selalu memberikan semangat dan menemani selama ini.
7. Teman-teman Program Studi Teknik Informatika khususnya angkatan 2011 kelas F yang selalu memberikan semangat dan kebersamaan selama penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
8. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah membantu dan terlibat secara langsung maupun tidak langsung dalam penulisan skripsi ini.

Hanya doa yang dapat penulisan berikan, semoga Allah SWT memberikan balasan yang setimpal atas jasa dan amal baik yang telah diberikan. Penulis menyadari bahwa skripsi ini tidak lepas dari kesalahan dan kekurangan. Oleh karena itu, penulis bersedia menerima kritik dan saran yang membangun untuk pengembangan penelitian selanjutnya. Penulis berharap semoga skripsi ini dapat memberikan manfaat.

## ABSTRAK

**Julita Gandasari Ariana, Rancang Bangun Aplikasi Pengelolaan Bahan Baku Kafe Menggunakan Metode *Analytical Hierarchy Process* (AHP) Dan *Push Notification***

**Pembimbing: Aryo Pinandito, S.T, M.MT, Muhammad Tanzil Furqon, S.Kom, M.CompSc**

Guna mempertahankan kualitas pelayanan, sebuah kafe harus memperhatikan beberapa aspek. Salah satunya adalah ketersediaan menu makanan atau minuman yang ditawarkan oleh kafe. Ketersediaan menu makanan atau minuman bergantung pada ketersediaan bahan baku penyusun makanan atau minuman. Maka dari itu manajemen kafe harus dapat memastikan bahwa suplai bahan baku mencukupi atau dapat dengan cepat disuplai ulang ketika dibutuhkan. *Push Notification* merupakan sebuah teknologi yang memungkinkan seseorang mendapatkan informasi melalui peringatan instan pada perangkat bergerak *smartphone*. Konsep ini dapat digunakan oleh manajemen kafe dalam melakukan *monitoring* informasi ketersediaan bahan baku, sehingga apabila terdapat bahan baku yang perlu disuplai ulang dapat ditangani dengan lebih cepat. Metode *Analytical Hierarchy Process* (AHP) digunakan untuk menentukan prioritas bahan baku yang harus disuplai ulang terlebih dahulu. Sisa bahan baku, jumlah menu yang menggunakan bahan baku, dan harga bahan baku menjadi kriteria yang diperhitungkan dalam menentukan prioritas suplai. Dengan menggabungkan kedua metode tersebut didapatkan hasil pengujian akurasi AHP sebesar 68%, dan tingkat keberhasilan pengiriman *push notification* sebesar 80%.

Kata kunci: *Analytical Hierarchy Process, Push Notification, Kafe, Suplai*

## ABSTRACT

**Julita Gandasari Ariana, Café Ingredients Management System Design Using Analytical Hierarchy Process (AHP) and Push Notification**

**Lecturer: Aryo Pinandito, S.T, M.MT, Muhammad Tanzil Furqon, S.Kom, M.CompSc**

To maintain quality of service, a café must pay attention to several aspect. One of them is the availability of the menu served by the café. The availability of the menu also depends on the availability of the ingredients needed to make it. Therefore, the café manager need to make sure that the ingredients is enough and resupplied fast enough when needed. *Push Notification* is a technology which allow information to be delivered to stakeholder via instant alert on their smartphone. This concept can be used by café manager to monitor the availability of ingredients, so that any insufficient ingredients can be resupplied fast. Analytical Hierarch Process (AHP) is used to determine the priority of ingredients that needed to be resupplied first. Ingredient stock, the number of menu using corresponding ingredient, and ingredient price are used to determine supply priority. The result of using both method reach 68% in accuracy, and 80% in delivery rate of push notification.

**Keyword:** *Analytical Hierarchy Process, Push Notification, Café, Supply*



## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT atas anugerah dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Aplikasi Pengelolaan Bahan Baku Kafe Menggunakan Metode *Analytical Hierarchy Process* (AHP) dan *Push Notification*” ini. Skripsi ini disusun sebagai syarat memperoleh gelar sarjana pada Program Studi Teknik Informatika, Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.

Dalam proses penyelesaian skripsi ini penulis mendapatkan banyak bantuan baik moral maupun materiil dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

9. Kedua orang tua, yaitu Drs. Arief Susanto Ak., CES dan Sri Harini beserta anggota keluarga yang lain yang telah memberikan dukungan dengan segala usaha dan doa selama ini.
10. Aryo Pinandito, S.T, M.MT selaku Pembimbing I yang telah memberikan bimbingan, ilmu, dan saran dalam penyelesaian skripsi ini.
11. Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku Pembimbing II yang juga telah memerikan bimbingan, ilmu, dan saran dalam penyelesaian skripsi ini.
12. Bapak dan Ibu Dosen yang telah mendidik dan memberikan ilmu selama penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
13. Ferdha Agisyanto selaku pemilik kafe Bunchbead yang telah memberikan data dan penjelasan yang diperlukan terkait kebutuhan skripsi ini.
14. Teman-teman terdekat yaitu Ahmad Dzulfikar S.R., Muhammad Murtadho, Rizky Kharisma, Stefanus Bayu Waskito, Andi Mohammad Rizki H., I Gusti Ngurah Ersania Susena, dan teman-teman yang lain yang membantu memberikan saran dalam pengerjaan dan selalu memberikan semangat dan menemani selama ini.
15. Teman-teman Program Studi Teknik Informatika khususnya angkatan 2011 kelas F yang selalu memberikan semangat dan kebersamaan selama penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
16. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah membantu dan terlibat secara langsung maupun tidak langsung dalam penulisan skripsi ini.

Hanya doa yang dapat penulisan berikan, semoga Allah SWT memberikan balasan yang setimpal atas jasa dan amal baik yang telah diberikan. Penulis menyadari bahwa skripsi ini tidak lepas dari kesalahan dan kekurangan. Oleh karena itu, penulis bersedia menerima kritik dan saran yang membangun untuk pengembangan penelitian selanjutnya. Penulis berharap semoga skripsi ini dapat memberikan manfaat.

Malang, 3 Agustus 2018

Penulis

arianajulita@gmail.com



## DAFTAR ISI

PENGESAHAN .....	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS .....	Error! Bookmark not defined.
KATA PENGANTAR .....	iii
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL .....	x
DAFTAR GAMBAR .....	xii
BAB 1 PENDAHULUAN .....	13
1.1 Latar belakang .....	13
1.2 Rumusan masalah .....	14
1.3 Tujuan .....	14
1.4 Manfaat .....	14
1.5 Batasan masalah .....	15
1.6 Sistematika pembahasan .....	15
BAB 2 LANDASAN KEPUSTAKAAN .....	16
2.1 Kajian pustaka .....	16
2.2 Kafe .....	16
2.3 <i>Analytical Hierarchy Process</i> .....	17
2.4 <i>Push Notification</i> .....	19
2.5 Firebase .....	19
2.5.1 Real-time database .....	19
2.5.2 Authentication .....	20
2.5.3 Cloud messaging .....	20
BAB 3 METODOLOGI .....	22
3.1 Studi literatur .....	22
3.2 Pengumpulan data .....	23
3.3 Analisis kebutuhan .....	23
3.4 Perancangan .....	23
3.5 Implementasi .....	23 ix

3.6 Pengujian .....	23
3.7 Pengambilan Kesimpulan dan Saran .....	24
BAB 4 PERANCANGAN .....	25
4.1 Analisis kebutuhan perangkat lunak .....	26
4.1.1 Gambaran umum sistem .....	26
4.1.2 Identifikasi aktor .....	27
4.1.3 Kebutuhan fungsional .....	27
4.1.4 Diagram <i>use case</i> .....	28
4.1.5 Skenario <i>use case</i> .....	28
4.1.6 Diagram <i>activity</i> .....	31
4.2 Perancangan .....	33
4.2.1 Perancangan arsitektural .....	33
4.2.2 Diagram <i>sequence</i> .....	34
4.2.3 Diagram <i>class</i> .....	35
4.2.4 Perancangan basis data .....	36
4.2.5 Desain antarmuka .....	36
4.2.6 Perancangan Algoritma AHP .....	38
4.2.7 Perancangan pengujian .....	45
BAB 5 IMPLEMENTASI .....	47
5.1 Batasan implementasi .....	47
5.2 Implementasi basis data .....	47
5.3 Implementasi kode program .....	48
BAB 6 PENGUJIAN DAN ANALISIS .....	59
6.1 Pengujian .....	59
6.2 Pengujian validasi .....	59
6.3 Pengujian akurasi .....	63
BAB 7 PENUTUP .....	68
7.1 Kesimpulan .....	68
7.2 Saran .....	68
DAFTAR PUSTAKA .....	69 x



## DAFTAR TABEL

Tabel 2.1 Skala Perbandingan Berpasangan .....	17
Tabel 2.2 Matriks Perbandingan Berpasangan .....	18
Tabel 2.3 Tabel Random Indeks .....	19
Tabel 4.1 Identifikasi Aktor .....	27
Tabel 4.2 Kebutuhan Fungsional Sistem .....	27
Tabel 4.3 Skenario <i>Use Case</i> Mengelola Pesanan .....	29
Tabel 4.4 Skenario <i>Use Case</i> Mengelola Bahan Baku .....	29
Tabel 4.5 Skenario <i>Use Case</i> Mengirim <i>Push Notification</i> .....	30
Tabel 4.6 Skenario <i>Use Case</i> Mengelola Bahan Baku .....	30
Tabel 4.7 Daftar Bahan Baku Yang Kurang Dari Batas Minimum .....	42
Tabel 4.8 Pengelompokan Data Kriteria Sisa .....	42
Tabel 4.9 Pengelompokan Data Kriteria Menu Yang Menggunakan .....	43
Tabel 4.10 Pengelompokan Data Kriteria Harga .....	43
Tabel 4.11 Hasil pengelompokan data alternatif .....	43
Tabel 4.12 Matriks perbandingan antar alternatif pada kriteria sisa bahan .....	44
Tabel 4.13 Normalisasi Matriks Perbandingan Antar Alternatif Kriteria Sisa .....	44
Tabel 4.14 Rata-rata Hasil Normalisasi .....	44
Tabel 4.15 Hasil akhir .....	45
Tabel 4.16 Tabel Pengujian Fungsional ( <i>Blackbox</i> ) .....	45
Tabel 5.1 Implementasi struktur basis data .....	47
Tabel 5.2 Kode implementasi AHP .....	48
Tabel 5.3 Implementasi <i>push notification</i> menggunakan Firebase Cloud Functions .....	54
Tabel 5.4 Kode program melihat <i>push notification</i> dan <i>update</i> bahan baku .....	55
Tabel 6.1 Kasus Uji Mengelola Pesanan .....	59
Tabel 6.2 Kasus Uji Mengelola Bahan Baku .....	59
Tabel 6.3 Kasus Uji Mengelola Bahan Baku (lanjutan).....	60
Tabel 6.4 Kasus Uji Mengirim <i>Push Notification</i> .....	60
Tabel 6.5 Kasus Uji Melihat <i>Push Notification</i> .....	60
Tabel 6.6 Kasus Uji Menambah Jumlah Bahan Baku .....	61 xi

Tabel 6.7 Kasus Uji Mengecek Perhitungan AHP .....	61
Tabel 6.8 Hasil Pengujian .....	62
Tabel 6.9 Tabel Perbandingan Data Sistem Dengan Data Kafe .....	63
Tabel 6.10 Perbandingan Pendapat Hari Pertama .....	64
Tabel 6.11 Perbandingan Pendapat Hari Kedua .....	64
Tabel 6.12 Perbandingan Pendapat Hari Ketiga .....	65
Tabel 6.13 Perbandingan Pendapat Hari Keempat .....	65



## BAB 1 PENDAHULUAN

Bab pendahuluan berisi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

### 1.1 Latar belakang

Kafe merupakan tempat yang pengunjungnya dapat memesan minuman atau makanan ringan dan biasanya digunakan sebagai tempat untuk berkumpul dengan teman maupun keluarga (Tanjung, 2013). Beberapa kafe memiliki menu unik yang menjadi andalan, agar kafe tersebut memiliki ciri khas dan tetap ramai. Bahan-bahan yang digunakan untuk menggunakan sebuah minuman atau makanan tentu harus benar-benar diperhatikan kualitasnya, jangan sampai ada bahan makanan yang busuk.

Sebagai contoh pada kafe Bunchbead, berdasarkan pernyataan pemilik kafe, pemilik kafe membeli bahan makanan setiap pagi dan beberapa dari bahan baku yang harus disediakan, dipesan dan diantar sebelum kafe dibuka. Pemilik kafe harus bisa memperkirakan bahan baku yang dibeli cukup untuk memenuhi menu yang akan dipesan oleh pelanggan pada hari itu. Jika bahan baku habis pada pertengahan hari saat dibukanya kafe dan pemilik kafe sedang tidak ada di tempat, tentu akan mengecewakan pelanggan.

*Push Notification* memperbolehkan sebuah aplikasi untuk memberitahu aplikasi lainnya bila ada peristiwa baru tanpa harus membuka aplikasi tersebut (Nations, n.d.). Dengan teknologi ini, diharapkan dapat memudahkan pemilik kafe dalam mendapatkan informasi mengenai bahan baku menu yang hampir habis di manapun pemilik kafe tersebut berada tanpa perlu memeriksanya dengan mendatangi kafe secara langsung. Pemilik kafe juga akan mendapatkan informasi mengenai bahan baku yang harus diprioritaskan untuk dibeli. Penentuan prioritas bahan pangan dapat ditentukan dengan menggunakan metode *Analytical Hierarchy Process* (AHP). AHP merupakan metode yang menguraikan masalah kompleks dan tidak terstruktur menjadi susunan hirarki yang memiliki nilai prioritas tentang pentingnya setiap variable (Saaty, 1999).

Hermawan Ardiyanto (2013), pada penelitiannya yang berjudul "Sistem Pendukung Keputusan Pemilihan Perumahan menggunakan Metode AHP berbasis Web (Studi Kasus CV. Elisma Anungkriya Demak)" menggunakan empat kriteria, yaitu harga, lokasi, tipe, dan jumlah.

Selain itu, Bagus Prasetyo, Wawan Laksito Y.S., dan Sri Siswanti (2013), melakukan penelitian yang berjudul "Sistem Pendukung Keputusan Pemilihan Paket Internet Operator Telekomunikasi dengan Metode AHP (Analytical Hierarchy Process)" memiliki sepuluh kriteria. Dalam metode AHP, terdapat fase pembobotan kriteria. Pembobotan kriteria ini bertujuan untuk mengetahui prioritas kriteria menurut pengguna. Hasil akhir perhitungan menggunakan metode AHP sangat bergantung pada pemberian bobot nilai terhadap kriteria

dan sub kriteria, sehingga hasil akhir prioritas dapat disesuaikan dengan kriteria yang benar-benar diprioritaskan.

AHP dapat diterapkan dengan baik, dengan contoh pada *“Halal Culinary Tracking Application at Food Souvenirs Center Based on Analytical Hierarchy Process (AHP) Method”* menghasilkan alternatif kemudahan akses sebagai rank pertama (Sucipto Sucipto, Mas'ud Effendi, Mochamad Ulul Khilmi, Muhammad Arif Kamal, Aryo Pinandito, Herman Tolle 2018). Selain itu, pada penelitian lain berjudul *“Alternatives weighting in analytic hierarchy process of mobile culinary recommendation system using fuzzy”*, metode AHP juga dapat dikombinasikan dengan metode lain dalam pembobotan sistem rekomendasi kuliner berbasis *mobile* (Aryo Pinandito, Mahardeka Tri Ananta, Komang Candra Brata, Lutfi Fanani 2015).

Oleh karena itu, untuk membantu pemilik kafe dalam pengelolaan bahan bakunya, diselesaikan dengan membangun aplikasi yang memiliki teknologi *Push Notification* untuk memberitahu pemilik kafe jika bahan baku kafanya sudah hampir habis dan metode AHP di dalamnya untuk penentuan prioritas bahan baku yang harus dibeli terlebih dahulu saat beberapa macam bahan baku kafe habis pada waktu yang bersamaan.

## 1.2 Rumusan masalah

Rumusan masalah penelitian yang dilakukan adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode AHP untuk membantu pemberian rekomendasi bahan baku yang harus dibeli terlebih dahulu?
2. Bagaimana struktur, rancangan dan implementasi *push notification* pada perangkat Android menggunakan Firebase?
3. Bagaimana tingkat akurasi dari hasil implementasi AHP pada sistem pengelolaan bahan baku kafe?

## 1.3 Tujuan

Tujuan dari penelitian yang dilakukan adalah untuk membuat aplikasi yang dapat segera memberi informasi kepada pemilik kafe mengenai kondisi stok bahan baku menggunakan teknik *push notification*, menerapkan dan mengetahui tingkat akurasi metode AHP untuk merekomendasikan prioritas bahan baku yang harus dibeli terlebih dahulu oleh pemilik kafe pada saat beberapa macam bahan baku kafe habis pada saat yang bersamaan.

## 1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian yang dilakukan adalah:

1. Memudahkan pemilik kafe untuk mengetahui stok bahan makanan yang hampir habis tanpa harus mengecek terlebih dahulu.
2. Memudahkan pemilik kafe untuk mengetahui bahan makanan yang diprioritaskan dalam pembeliannya.



## 1.5 Batasan masalah

Batasan masalah pada penelitian yang dilakukan adalah:

1. Lokasi penelitian berada pada kafe Bunchbead.
2. Aplikasi dibuat hanya meliputi pengelolaan bahan baku kafe.

## 1.6 Sistematika pembahasan

Sistematika pembahasan memberikan gambaran dan uraian penulisan secara singkat.

### 1. Bab 1 Pendahuluan

Memuat latar belakang penelitian, rumusan masalah tentang beberapa masalah yang ingin diselesaikan, tujuan dari penelitian, manfaat penelitian, batasan masalah agar penelitian ini tidak melebar pada bidang lain dan fokus penelitian tetap pada masalah yang ingin diselesaikan, dan sistematika pembahasan.

### 2. Bab 2 Landasan Kepustakaan

Berisikan penjelasan dasar teori dari berbagai sumber pustaka dan memuat penelitian yang sudah ada sebelumnya untuk dijadikan referensi dan menunjang penyelesaian masalah pengelolaan bahan baku kafe.

### 3. Bab 3 Metodologi

Bab metodologi membahas mengenai metode yang akan dilakukan untuk menyelesaikan permasalahan yang ada pada rumusan masalah dan metode-metode yang dilakukan dalam pengerjaan skripsi.

### 4. Bab 4 Perancangan

Bab perancangan membahas mengenai analisis kebutuhan dan perancangan sistem.

### 5. Bab 5 Implementasi

Membahas tentang implementasi sistem berdasarkan analisis kebutuhan dan perancangan dari sistem yang telah dibuat.

### 6. Bab 6 Pengujian

Memuat hasil pengujian dan analisis sistem yang telah direalisasikan, berdasarkan perancangan sistem.

### 7. Bab 7 Penutup

Memuat kesimpulan beserta saran yang diperoleh berdasarkan rumusan masalah yang ada, proses perancangan, implementasi dan pengujian sistem untuk pengembangan lebih lanjut.

## BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan berisi kajian pustaka dan dasar teori yang berhubungan dengan pengembangan perangkat lunak yang dilakukan. Kajian pustaka berupa pembahasan mengenai penelitian yang telah ada sebelumnya. Dasar teori yang dimaksud adalah kafe, AHP, *Push Notification*, dan Firebase.

### 2.1 Kajian pustaka

Terdapat dua sampel penelitian, yaitu Sistem Pendukung Keputusan Pemilihan Perumahan Menggunakan Metode AHP Berbasis Web (Ardiyanto, 2013) dan Sistem Pendukung Keputusan Pemilihan Paket Internet Operator Telekomunikasi dengan Metode AHP (*Analytical Hierarchy Process*) (Bagus Prasetyo, 2013). Dalam dua penelitian tersebut dilakukan penelitian dengan metode yang sama, yaitu metode AHP dan secara global meneliti tentang sistem yang mampu merekomendasi terhadap pemilihan sesuatu.

Meskipun menggunakan metode yang sama, tetapi dua penelitian tersebut akan memberikan hasil yang berbeda dengan penelitian yang akan dilakukan, karena topik yang diambil berbeda. Dengan berbedanya topik yang diambil, tentu memiliki input, parameter, dan output yang berbeda, misalnya pada penelitian Sistem Pendukung Keputusan Pemilihan Perumahan Menggunakan Metode AHP Berbasis Web, memiliki parameter harga, lokasi, dan tipe. Sedangkan pada penelitian Sistem Pendukung Keputusan Pemilihan Paket Internet Operator Telekomunikasi dengan Metode AHP (*Analytical Hierarchy Process*) memiliki parameter biaya kehidupan, biaya kecepatan, biaya kuota, biaya paket, kecepatan kuota, kecepatan paket dan kuota paket.

Output yang dihasilkan oleh kedua penelitian tersebut juga berbeda. Pada penelitian Sistem Pendukung Keputusan Pemilihan Perumahan menggunakan Metode AHP memiliki output ranking pemilihan perumahan untuk konsumen beserta nilai kriteria dari masing-masing perumahan. Perumahan yang disarankan memiliki ranking tertinggi, yang berarti perumahan tersebut memiliki nilai kriteria tertinggi dibandingkan dengan perumahan lainnya. Output pada penelitian Sistem Pendukung Keputusan Pemilihan Paket Internet Operator Telekomunikasi dengan Metode AHP (*Analytical Hierarchy Process*) adalah menampilkan satu nama provider internet yang memiliki nilai kriteria tertinggi berdasarkan tujuh kriteria yang ada, yaitu biaya kebutuhan, biaya kecepatan, biaya kuota, biaya paket, kecepatan kuota, kecepatan paket, dan kuota paket.

### 2.2 Kafe

Kafe merupakan tempat yang melayani pesanan minuman hangat. Kafe mirip dengan restoran atau bar, tapi berbeda dengan kafetaria. Kafe masa kini banyak yang menyediakan minuman lain seperti minuman dingin dan makanan ringan. Kafe kini telah menjadi tempat untuk membaca, menulis, berkumpul dengan teman, maupun sekedar mengisi waktu. (Tanjung, 2013)

Lokasi kafe berada di pusat perbelanjaan atau di ruang publik yang ramai. Selain makanan, sebagian kafe juga menyediakan penampilan musik sebagai hiburan bagi pengunjung. Dekorasi kafe dibuat menarik untuk kenyamanan pengunjung dan untuk menarik minat pengunjung dan bisa memberikan kesan tersendiri. (Maulidi, 2017)

### 2.3 Analytical Hierarchy Process

AHP digunakan untuk membuat sebuah model permasalahan yang tidak terstruktur dengan menggunakan pendapat orang yang memiliki hubungan erat dengan permasalahan yang diangkat dan memiliki skala preferensi yang telah didefinisikan. Skala preferensi dijabarkan pada Tabel 2.1.

**Tabel 2.1 Skala Perbandingan Berpasangan**

Tingkat Kepentingan	Definisi
1	Kedua elemen sama penting
3	Elemen pertama sedikit lebih penting daripada elemen kedua
5	Elemen pertama lebih penting daripada elemen kedua
7	Elemen pertama jelas lebih penting daripada elemen kedua
9	Elemen pertama mutlak lebih penting daripada elemen kedua
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Bila nilai aktivitas pertama terhadap aktivitas kedua telah didapatkan, maka nilai dari aktivitas kedua terhadap aktivitas pertama adalah kebalikannya

Sumber: diadaptasi dari (Saaty, 1999)

Skala perbandingan berpasangan digunakan untuk memberi nilai pada perbandingan dua variabel pada tingkat tertentu seperti kriteria, lebih penting atau kurang penting. Hasil penilaian disajikan dalam matriks perbandingan berpasangan. Orang yang berhak untuk memberi nilai merupakan orang yang terlibat langsung dengan permasalahan yang ada (Saaty, 1999). Matriks perbandingan berpasangan digambarkan pada Tabel 2.2.  $C_1$ ,  $C_2$  hingga  $C_n$  merupakan kriteria yang ditentukan. Nilai  $a$  merupakan nilai pendapat individu. Nilai yang diletakkan pada  $a_{12}$  merupakan nilai pendapat yang didapat dari perbandingan kriteria pertama terhadap kriteria kedua. Bila nilai tersebut adalah 3, maka kriteria pertama sedikit lebih penting bila dibandingkan dengan kriteria kedua. Bila diberi nilai 9, maka kriteria pertama mutlak lebih penting bila dibandingkan dengan kriteria kedua. Nilai  $1/a_{12}$  merupakan kebalikan dari  $a_{12}$ , sehingga nilai  $1/a_{12}$  adalah 1 dibagi 3.

**Tabel 2.2 Matriks Perbandingan Berpasangan**

	C1	C2	...	Cn
C1	1	a <sub>12</sub>	...	a <sub>1n</sub>
C2	1/a <sub>12</sub>	1	...	a <sub>2n</sub>
...	...	...	...	...
Cn	1/a <sub>1n</sub>	1/a <sub>2n</sub>	...	1

Sumber: diadaptasi dari (Saaty, 1999)

Langkah selanjutnya adalah menormalisasi matriks perbandingan berpasangan. Normalisasi dilakukan agar jumlah nilai setiap kolom adalah 1. Nilai matriks normalisasi  $B = [b_{ij}]$  didapatkan dari membagi nilai dari kolom matriks  $A = [a_{ij}]$  atau disebut juga matriks perbandingan berpasangan, dengan jumlah pada masing-masing kolomnya, dengan rumus:

$$b_{ij} = \frac{a_{ij}}{\sum_{i=1}^n a_{ij}} \quad (2.1)$$

Pembobotan tiap kriteria dilakukan dengan membagi jumlah baris dengan jumlah banyaknya kriteria yang ada, dengan rumus:

$$C_{i1} = \frac{1}{n} \sum_{j=1}^n b_{ij} \quad (2.2)$$

Langkah selanjutnya adalah mengecek konsistensi matriks pendapat individu yang telah dibuat. Matriks pendapat individu konsisten jika nilai CI kurang dari atau sama dengan 0.1 (Saaty, 1999). Terdapat lima langkah untuk mengecek konsistensi matriks pendapat individu, yaitu:

1. Menghitung vektor jumlah pembobotan dengan mengalikan matriks pendapat individu dan bobot prioritas, dengan rumus:

$$w_{i1} = \sum_{j=1}^n a_{ij} C_{j1} \quad (2.3)$$

2. Menghitung vektor konsistensi dengan cara membagi masing-masing nilai vektor jumlah pembobotan dengan bobot prioritas, dengan rumus:

$$c_{i1} = \frac{w_{i1}}{C_{i1}} \quad (2.4)$$

3. Menghitung lambda maksimum ( $\lambda_{\max}$ ) dengan menghitung rata-rata dari vektor konsistensi.
4. Menghitung indeks konsistensi dengan cara mencari nilai CI terlebih dahulu yaitu membagi nilai dari lambda maksimum dikurangi jumlah kriteria dan nilai jumlah kriteria dikurangi satu, dengan rumus:



$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (2.5)$$

5. Menghitung *Consistency Ratio (CR)*. Nilai CR didapat dari membagi nilai CI dan nilai RC, rumus:

$$CR = \frac{CI}{RC} \quad (2.6)$$

RC merupakan nilai yang didapatkan dari tabel random indeks. Saat  $CR < 0.1$ , maka nilai perbandingan berpasangan adalah konsisten. Jika nilai perbandingan berpasangan tidak konsisten, maka nilai pada matriks berpasangan harus diulang (Saaty, 1999). Nilai random indeks dijabarkan pada Tabel 2.3.

**Tabel 2.3 Tabel Random Indeks**

Ukuran matriks (n)	1	2	3	4	5	6	7	8	9	10	11	12
Random indeks (RI)	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.58

Sumber: (Ariff, Salit, Ismail, & Nukman, 2008)

## 2.4 Push Notification

*Push* merupakan pemberitahuan yang dilakukan oleh server sehingga dapat memberitahu *client* tanpa harus membuka aplikasi setiap beberapa saat. Contohnya, penerima *email* akan mengerti bila ada *email* yang baru masuk ke kotak masuknya tepat setelah *email* tersebut dikirim. Metode ini pertama kali dipopulerkan oleh *Blackberry*. Jalur yang digunakan untuk *push* pada android pertama kali melalui C2DM (Random Log, 2010).

## 2.5 Firebase

Firebase merupakan *framework* yang berguna untuk membuat aplikasi gawai maupun aplikasi web yang membutuhkan akses database secara *real-time*, sehingga ketika seorang pengguna mengubah data pada aplikasi, setiap pengguna lain akan langsung mengetahui perubahan data tersebut. Firebase memegang banyak peranan dari sisi server dalam pengembangan sebuah aplikasi. Firebase memberikan banyak fitur yang berkaitan dengan produk-produk Google lainnya (Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta, Dr. Asoke Nath, 2018).

### 2.5.1 Real-time database

Data yang disimpan pada real-time database dalam bentuk JSON, dan disinkronisasikan terus menerus kepada setiap klien. Klien dapat memiliki aplikasi dari beberapa platform yang berbeda, seperti iOS, Android, Javascript. Pengembangan aplikasi dengan platform yang berbeda tersebut menjadi lebih mudah, karena database yang telah dibuat bisa digunakan untuk banyak

platform sekaligus, dan Firebase telah memegang kebanyakan proses *back-end* dari aplikasi (Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta, Dr. Asoke Nath, 2018). Gambar 2.1 merupakan contoh data JSON yang digunakan pada Firebase Real-time Database. JSON memiliki struktur yang berbeda dari database SQL, karena dibuat sedatar mungkin dan tidak perlu adanya normalisasi data.

```
{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      "contacts": { "ghopper": true },
    },
    "ghopper": { ... },
    "eclarke": { ... }
  }
}
```

**Gambar 2.1 Struktur JSON**

Sumber: (Developers, Firebase, 2018)

### 2.5.2 Authentication

Firebase Authentucation adalah salah satu fitur Firebase yang menyediakan pengaturan akses pengguna terhadap aplikasi yang dibuat oleh pengembang. Firebase menyediakan login melalui Gmail, Github, Twitter, Facebook, dan juga membiarkan pengembang untuk membuat otentikasinya sendiri. (Sonam Khedkar, Swapnil Thube, 2017)

### 2.5.3 Cloud messaging

Cloud Messaging merupakan solusi bagi pengembang yang menggunakan banyak platform dan ingin memberi pesan kepada pengguna tanpa mengeluarkan biaya. Hal pertama yang dilakukan untuk dapat menggunakan Cloud Messaging adalah mendaftarkan masing-masing perangkat pada server Cloud Messaging. Pada server, pengembang dapat memasukkan berita yang ingin disampaikan pada masing-masing pengguna, dan Firebase akan mengirimkan pesan tersebut kepada pengguna dengan menggunakan ID yang telah diberikan pengembang di server, seperti pada Gambar 2.2.



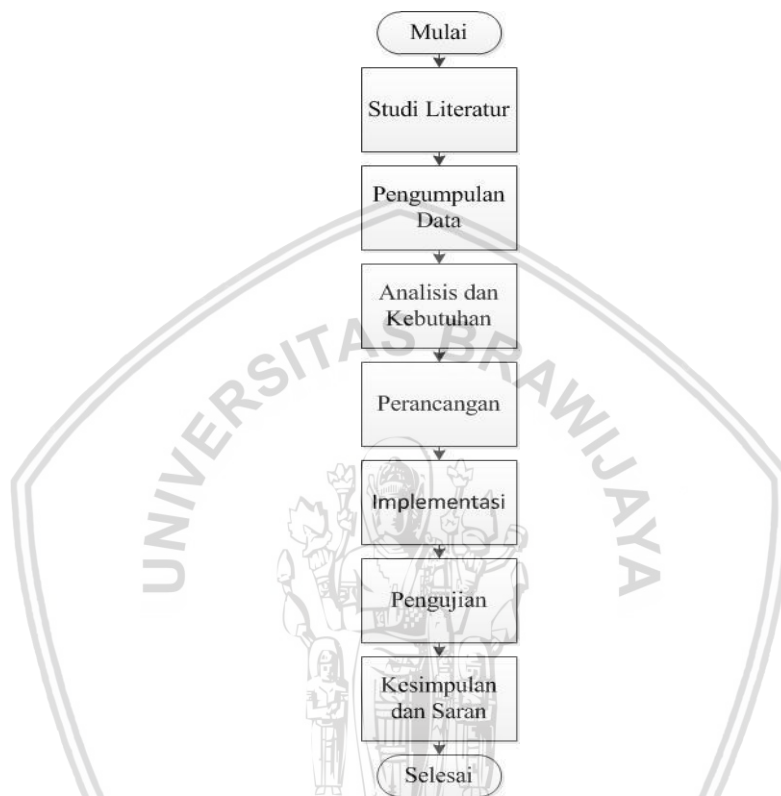
**Gambar 2.2 Firebase Cloud Messaging**

Sumber: (Developers, Firebase, 2018)

Pada Gambar 2.2 dijelaskan mengenai interaksi antara Firebase Cloud Messaging Server, aplikasi server, dan aplikasi klien. Aplikasi server atau Notification Console GUI mengirim data kepada Firebase Cloud Messaging lalu meneruskannya ke aplikasi klien. Aplikasi klien berupa perangkat Android, web, maupun iOS.

## BAB 3 METODOLOGI

Bab metodologi berisi metode penelitian yang dilakukan pada penelitian, untuk menerapkan metode AHP dan *Push Notification*. Metode penelitian yang dilakukan meliputi studi literatur, pengumpulan data, analisis kebutuhan, perancangan, implementasi, pengujian, serta menentukan kesimpulan dan memberikan saran, yang digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

### 3.1 Studi literatur

Studi literatur merupakan pengumpulan informasi atau data-data terpercaya yang diperlukan dengan sumber berupa buku, e-book, jurnal/paper, dan penelitian sebelumnya. Informasi tersebut dijadikan dasar teori dan kajian pustaka yang akan diimplementasikan dalam penelitian.

Studi literatur yang digunakan adalah:

1. Kafe
2. Analytical Hierarchy Process
3. Push Notification
4. Firebase



### 3.2 Pengumpulan data

Pengumpulan data dilakukan setelah studi literatur dilakukan. Data yang digunakan didapat dari kafe Bunchbead, berupa macam-macam bahan, jumlah bahan yang tersedia, menu yang ada dan bahan-bahan yang digunakan sebagai menu tersebut.

### 3.3 Analisis kebutuhan

Analisis kebutuhan dilakukan untuk mendefinisikan hal-hal yang dibutuhkan untuk perancangan dan implementasi penelitian agar aplikasi dapat berjalan dengan baik sesuai dengan yang dibutuhkan. Analisis kebutuhan berasal dari identifikasi kebutuhan yang ingin diterapkan pada sistem. Identifikasi kebutuhan didapatkan dari hasil wawancara yang dilakukan dengan pemilik kafe Bunchbead. Analisis kebutuhan sistem dikemas dalam bentuk analisis aktor, diagram use case, dan diagram activity.

### 3.4 Perancangan

Setelah semua kebutuhan dijabarkan, maka dapat dilakukan perancangan pembuatan aplikasi sistem. Tahap perancangan dibuat berdasarkan analisis aktor, diagram *use case*, dan diagram *activity* yang ada. Perancangan dilakukan untuk memudahkan implementasi sistem yang akan dilakukan. Perancangan sistem membahas mengenai diagram *sequence*, diagram *class*, diagram *object*, perancangan *user interface*, dan perhitungan manual data menggunakan metode AHP.

### 3.5 Implementasi

Implementasi dilakukan berdasarkan pada perancangan yang telah dibuat. Aplikasi akan dibuat menjadi apk (Android Package) dengan menggunakan ADT (Android Developer Tools), yang menggunakan Firebase Cloud Messaging untuk membantu pengiriman data dari server ke aplikasi android. Selain *push notification* yang menggunakan Firebase Cloud Messaging, AHP juga diimplementasikan untuk membantu mengurutkan prioritas bahan yang harus dibeli oleh pemilik kafe pada saat itu.

### 3.6 Pengujian

Setelah implementasi pada perangkat lunak selesai, dilakukan pengujian pada perangkat lunak tersebut. Pengujian dilakukan untuk mengetahui bila implementasi yang dibuat telah sesuai dengan perancangan dan kebutuhan yang telah ada. Pengujian yang dilakukan adalah:

Pengujian *blackbox* dilakukan untuk mengetahui kesesuaian fungsionalitas aplikasi yang dibuat dengan perancangan yang ada. Pengujian *blackbox* dilakukan dengan menjalankan aplikasi berdasarkan dengan kebutuhan fungsional. Hasil pengujian akan bernilai valid bila sistem telah berjalan sesuai dengan kebutuhan

fungsional dan bernilai tidak valid bila sistem tidak berjalan sesuai dengan kebutuhan fungsional. Termasuk di dalamnya dilakukan pengujian validasi *push notification* dilakukan untuk mengetahui apakah *push notification* dikirim pada saat yang seharusnya.

Pengujian kedua adalah pengujian akurasi yang membandingkan hasil prioritas perhitungan sistem dengan prioritas pendapat pemilik kafe. Bila urutan prioritas sama, maka bernilai valid. Akurasi didapat dalam bentuk persentase.

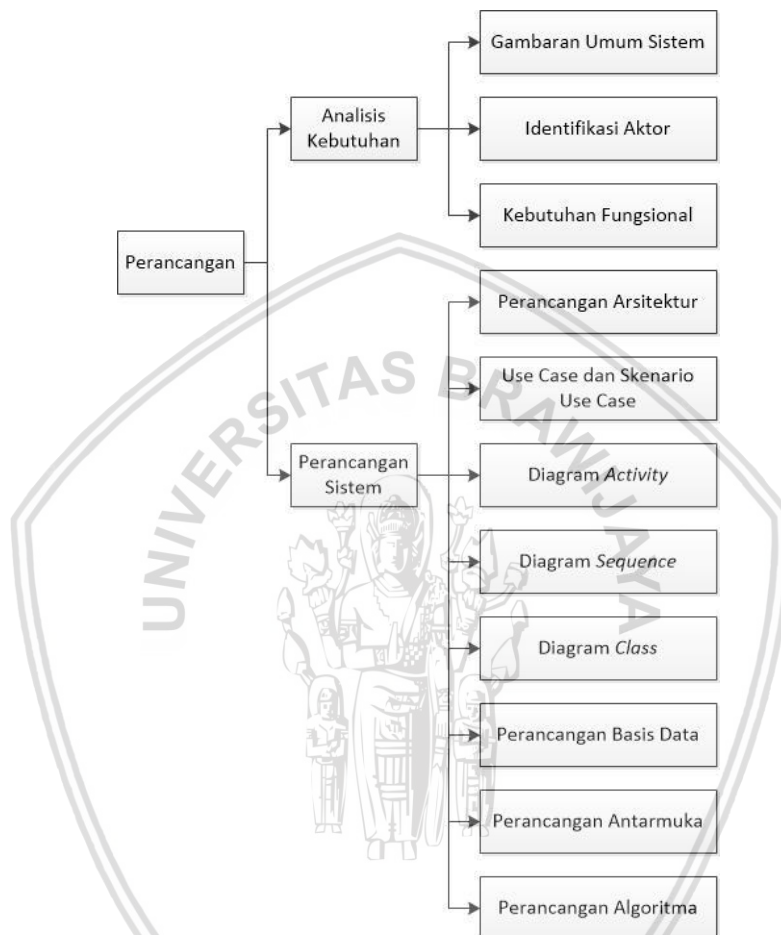
### 3.7 Pengambilan Kesimpulan dan Saran

Setelah pengumpulan data, analisis kebutuhan, perancangan, implementasi, dan pengujian, dilakukan pengambilan kesimpulan dan pemberian saran sebagai tahap terakhir yang dilakukan. Pengambilan kesimpulan dilakukan berdasarkan hasil pengujian sistem yang dibuat, apakah dapat menyelesaikan permasalahan yang ada pada rumusan masalah. Setelah pengambilan keputusan, penulisan saran dilakukan.



## BAB 4 PERANCANGAN

Pada bab perancangan akan dibahas mengenai analisis kebutuhan perangkat lunak dan perancangan perangkat lunak yang akan diimplementasikan. Pohon perancangan aplikasi pengelolaan bahan baku kafe menggunakan metode AHP berbasis android digambarkan dalam Gambar 4.1.



**Gambar 4.1 Pohon Perancangan Sistem**

Gambar 4.1 merupakan pohon perancangan dari perancangan sistem pengelolaan bahan baku menggunakan metode AHP dan *push notification*. Perancangan dibagi menjadi dua, yaitu analisis kebutuhan perangkat lunak dan perancangan sistem. Analisis kebutuhan perangkat lunak terdiri dari gambaran umum sistem, identifikasi aktor, dan kebutuhan fungsional. Pada perancangan sistem terdapat perancangan arsitektur sistem, diagram *use case*, skenario *use case*, diagram *activity*, diagram *sequence*, diagram *class*, perancangan basis data, desain antarmuka, dan perancangan algoritma.

## 4.1 Analisis kebutuhan perangkat lunak

Analisis kebutuhan perangkat lunak dibuat agar sistem yang dibuat dapat sesuai dengan kebutuhan pengguna. Analisis kebutuhan perangkat lunak meliputi gambaran umum sistem, identifikasi aktor, dan kebutuhan fungsional.

### 4.1.1 Gambaran umum sistem

Dalam gambaran umum sistem terdapat tiga bagian, yaitu deskripsi sistem, lingkungan sistem, dan proses bisnis sistem.

#### 4.1.1.1 Deskripsi sistem

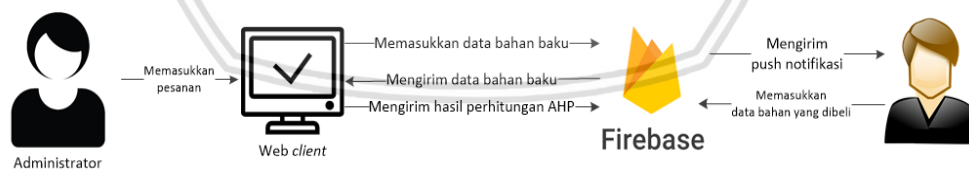
Sistem pengelolaan bahan baku kafe ini dibuat untuk memudahkan pemilik kafe mengetahui bahan baku yang hampir habis dan dapat segera membeli bahan baku tersebut. Bahan baku yang hampir habis secara bersamaan dihitung menggunakan AHP untuk merekomendasi bahan baku yang harus dibeli terlebih dahulu. Hasil perhitungan AHP tersebut dikirim menggunakan *push notification* kepada pemilik kafe, sehingga pemilik kafe dapat segera membeli bahan baku yang dibutuhkan. Setelah membeli bahan baku yang ada pada rekomendasi, pemilik kafe dapat menambah jumlah bahan baku yang baru dibeli ke dalam sistem.

#### 4.1.1.2 Lingkungan sistem

Sistem pengelolaan bahan baku kafe ini merupakan sistem yang terdiri dari dua bagian, yaitu aplikasi web untuk administrator dan aplikasi *mobile* untuk pemilik kafe. Aplikasi untuk administrator merupakan aplikasi web, yang digunakan untuk manajemen data bahan baku. Aplikasi untuk pemilik kafe adalah aplikasi *mobile*.

#### 4.1.1.3 Bisnis proses sistem

Gambaran umum bisnis proses ditunjukkan pada Gambar 4.2.



**Gambar 4.2 Bisnis Proses Sistem**

Langkah pertama bisnis proses ini adalah administrator memasukkan pesanan ke *web-client* untuk diolah, sehingga didapatkan data bahan baku yang akan digunakan dalam membuat menu yang telah dipesan. Proses kedua adalah melakukan *update* data bahan baku yang telah diolah. Data bahan baku yang telah diolah berfungsi untuk mengurangi jumlah bahan baku yang ada pada *database*. *Database* yang digunakan merupakan salah satu produk Firebase, yaitu Firebase Real-time Database. Setelah jumlah bahan baku berkurang, *web-client* mengambil data bahan baku dari *database* dan menghitungnya dengan menggunakan AHP. Setelah dihitung, hasil rekomendasi dari perhitungan AHP

dimasukkan ke *database*. Kemudian Firebase akan mengirim *push notification* kepada pemilik kafe yang berisi hasil rekomendasi perhitungan AHP. Setelah menerima *push notification*, pemilik kafe dapat menambahkan data bahan baku yang baru saja dibeli melalui aplikasi *mobile*.

#### 4.1.2 Identifikasi aktor

Aktor adalah orang yang akan menggunakan dan berhubungan dengan sistem ini. Terdapat dua aktor yang ada, yaitu pemilik kafe dan administrator, yang dijelaskan pada Tabel 4.1.

**Tabel 4.1 Identifikasi Aktor**

Aktor	Deskripsi
Administrator	Administrator bertugas untuk melakukan pengelolaan bahan baku. Administrator juga dapat mengirim <i>push notification</i> kepada pemilik kafe bila perhitungan AHP pada <i>web-client</i> telah memberikan hasil berupa bahan yang sudah hampir habis.
Pemilik kafe	Pemilik kafe menggunakan aplikasi <i>mobile</i> , dan akan menerima <i>push notification</i> yang telah dikirimkan yang berisi mengenai bahan baku yang hampir habis. Pemilik kafe juga dapat menambahkan data bahan baku yang telah dibeli berdasarkan rekomendasi.

#### 4.1.3 Kebutuhan fungsional

Kebutuhan fungsional sistem terdiri dari 5 kebutuhan yang telah dijelaskan oleh pemilik kafe Bunchbead. Kebutuhan-kebutuhan ini nantinya akan dimodelkan dalam bentuk *use case*. Deskripsi masing-masing kebutuhan dijelaskan pada Tabel 4.2.

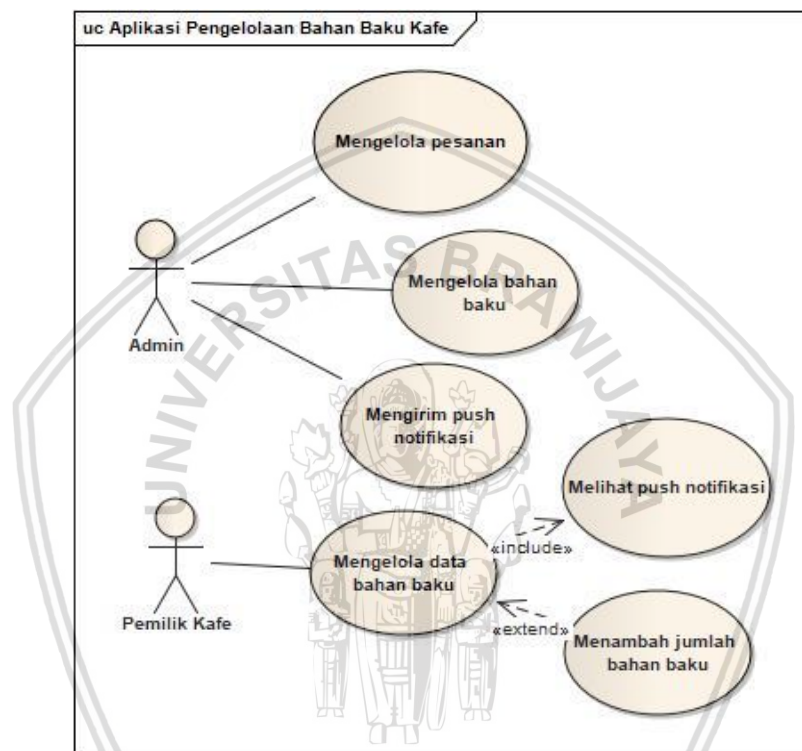
**Tabel 4.2 Kebutuhan Fungsional Sistem**

Nomor	Kebutuhan	Use Case
SRS-01	Administrator dapat mengelola bahan baku yang dibutuhkan	Mengelola daftar bahan baku
SRS-02	Administrator dapat mengelola menu-menu yang terjual tiap harinya	Mengelola pesanan
SRS-03	Administrator mengirim <i>push notification</i> kepada pemilik kafe berupa rekomendasi data yang telah dihitung oleh sistem menggunakan metode AHP	Mengirim <i>push notification</i>
SRS-04	Pemilik dapat melihat daftar bahan baku yang dikirimkan melalui <i>push notification</i>	Melihat <i>push notification</i>
SRS-05	Pemilik dapat melakukan pengubahan data bahan baku yang ada pada aplikasi <i>mobile</i> bila bahan baku sudah dibeli	Menambah jumlah bahan baku



#### 4.1.4 Diagram use case

Diagram *use case* dibuat berdasarkan kebutuhan fungsional yang telah dideskripsikan. Gambar 4.3 merupakan gambaran kebutuhan sistem secara singkat. Terdapat lima *use case* yang ada pada aplikasi pengelolaan bahan baku kafe, yaitu mengelola pesanan, mengelola bahan baku, dan mengirim *push notification* yang bisa dilakukan oleh administrator, kemudian melihat mengelola bahan baku dapat dilakukan oleh pemilik kafe. *Use case* mengelola data bahan baku *include* melihat *push notification*, dan *use case* menambah jumlah bahan baku adalah *exclude* dari mengelola data bahan baku.



Gambar 4.3 Diagram Use Case

#### 4.1.5 Skenario use case

Setiap *use case* memiliki tujuan, deskripsi, aktor yang menjalankan, dan proses skenario yang berbeda dengan yang lainnya. Proses skenario pada skenario *use case* menggambarkan langkah-langkah yang dilakukan oleh user pada sistem, pada *use case* tertentu. Penjelasan lebih lengkap mengenai masing-masing *use case* dijelaskan pada Tabel 4.3, Tabel 4.4, dan Tabel 4.5, dan Tabel 4.6.

Tabel 4.3 berisi skenario *use case* mengelola pesanan yang dilakukan oleh administrator. Administrator dapat menambahkan pesanan ke sistem dengan menekan tombol *new item*, sistem akan menampilkan formulir pesanan. Setelah administrator mengisi pesanan pada formulir yang ada dan menekan tombol *save*, sistem akan memasukkan data tersebut ke *database*.

Tabel 4.4 berisi skenario *use case* mengelola data bahan baku yang dilakukan oleh administrator. Kondisi awal adalah administrator telah berada pada halaman bahan baku. Kemudian administrator menekan tombol tambah bahan dan sistem akan menampilkan formulir tambah bahan baku. Setelah itu administrator mengisi formulir dan menekan tombol simpan, dan sistem akan memasukkan data yang ada pada formulir ke dalam *database*.

**Tabel 4.3 Skenario *Use Case* Mengelola Pesanan**

Identifikasi	
<b>Nama Use Case</b>	Mengelola pesanan
<b>Tujuan</b>	Administrator dapat mengelola daftar pesanan
<b>Deskripsi</b>	Administrator menambah daftar pesanan melalui <i>web-client</i>
<b>Aktor</b>	Administrator
<b>Proses Skenario: Mengelola pesanan</b>	
<b>Kondisi Awal</b>	Administrator berada pada halaman pesanan
<b>Aksi dari Aktor</b>	<b>Reaksi dari Sistem</b>
Administrator menekan tombol <i>new item</i>	<i>Web-client</i> menampilkan formulir penambahan pesanan
Administrator memasukkan pesanan ke formulir dan menekan tombol <i>save</i>	Sistem memasukkan data pesanan ke <i>database</i>
<b>Kondisi Akhir</b>	Sistem memasukkan data pesanan ke <i>database</i>

**Tabel 4.4 Skenario *Use Case* Mengelola Bahan Baku**

Identifikasi	
<b>Nama Use Case</b>	Mengelola bahan baku
<b>Tujuan</b>	Administrator dapat menambahkan data bahan baku baru
<b>Deskripsi</b>	Administrator menambahkan data bahan baku baru melalui <i>web-client</i>
<b>Aktor</b>	Administrator
<b>Proses Skenario: Mengelola bahan baku</b>	
<b>Kondisi Awal</b>	Administrator berada pada halaman bahan baku
<b>Aksi dari Aktor</b>	<b>Reaksi dari Sistem</b>
Administrator menekan tombol tambah bahan	Sistem menampilkan formulir tambah bahan baku
Administrator mengisi formulir dan menekan tombol simpan	Sistem memasukkan data yang ada pada formulir ke dalam <i>database</i>
<b>Kondisi Akhir</b>	Sistem mengirim <i>push notification</i> kepada aplikasi mobile pemilik kafe

Tabel 4.5 merupakan skenario dari *use case* mengirim *push notification* yang dilakukan oleh administrator. Administrator memasukkan pesanan ke *web-client*. Setelah itu, administrator menekan tombol *apply* jika ingin mengirim *push notification* ke perangkat Android pemilik kafe. *Push notification* berisi rekomendasi prioritas pembelian bahan baku hasil perhitungan menggunakan AHP.

**Tabel 4.5 Skenario Use Case Mengirim Push Notification**

Identifikasi	
<b>Nama Use Case</b>	Mengirim <i>push notification</i>
<b>Tujuan</b>	Administrator dapat mengirim <i>push notification</i> kepada pemilik kafe
<b>Deskripsi</b>	Administrator mengirim <i>push notification</i> kepada pemilik kafe berisi data yang telah dihitung oleh sistem menggunakan metode AHP
<b>Aktor</b>	Administrator
<b>Proses Skenario: Mengirim Push Notification</b>	
<b>Kondisi Awal</b>	Administrator berada pada halaman pesanan
<b>Aksi dari Aktor</b>	<b>Reaksi dari Sistem</b>
Administrator memasukkan pesanan	Sistem menambah daftar data pesanan
Administrator menekan tombol apply	Sistem menghitung bahan baku yang kurang dari minimum menggunakan metode AHP, kemudian mengirim <i>push notification</i> kepada aplikasi <i>mobile</i> pemilik kafe yang berisi rekomendasi bahan baku yang harus dibeli
<b>Kondisi Akhir</b>	Sistem mengirim <i>push notification</i> kepada aplikasi <i>mobile</i> pemilik kafe

**Tabel 4.6 Skenario Use Case Mengelola Bahan Baku**

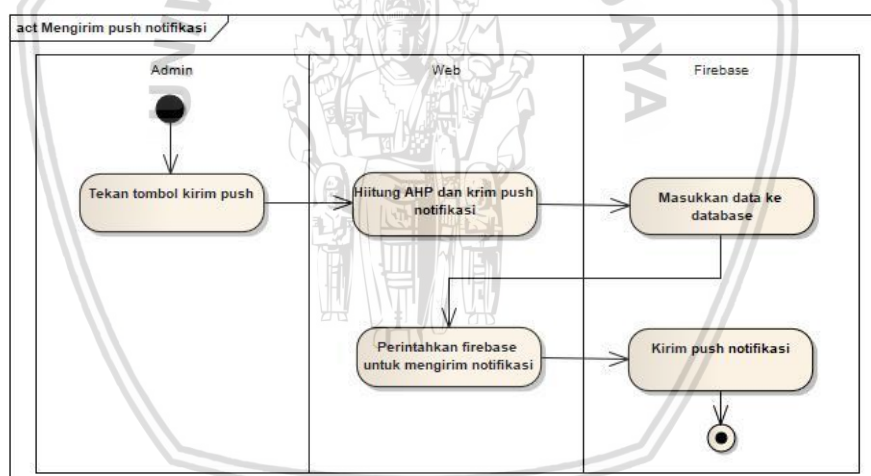
Identifikasi	
<b>Nama Use Case</b>	Mengelola data bahan baku
<b>Tujuan</b>	Mengelola data bahan baku berdasarkan <i>push notification</i> yang dikirimkan oleh administrator
<b>Deskripsi</b>	Pemilik menerima <i>push notification</i> dari administrator dan dapat menindaklanjuti <i>push notification</i> tersebut
<b>Aktor</b>	Pemilik kafe
<b>Proses Skenario: Melihat push notification</b>	
<b>Kondisi Awal</b>	Pemilik mendapatkan <i>push notification</i> dari sistem
<b>Aksi dari Aktor</b>	<b>Reaksi dari Sistem</b>
Pemilik membuka halaman <i>push notification</i>	Sistem menampilkan daftar rekomendasi bahan baku yang harus dibeli
<b>Kondisi Akhir</b>	Sistem menampilkan rekomendasi bahan baku yang harus dibeli
<b>Proses Skenario: Menambah jumlah bahan baku</b>	
<b>Kondisi Awal</b>	Pemilik berada pada halaman notifikasi

Aksi dari Aktor	Reaksi dari Sistem
Pemilik menekan salah satu bahan baku hasil rekomendasi	Sistem menampilkan data bahan baku tersebut beserta formulir penambahan bahan baku
Pemilik menekan tombol simpan	Sistem menyimpan data tambahan jumlah bahan baku ke <i>database</i>
Kondisi Akhir	Sistem menyimpan data tambahan jumlah bahan baku ke <i>database</i>

Tabel 4.6 merupakan aksi yang dilakukan oleh pemilik kafe setelah menerima *push notification* dari administrator. Pemilik dapat melihat rekomendasi daftar bahan baku yang harus dibeli terlebih dahulu, lalu menambah bahan baku setelah bahan tersebut dibeli. Sistem akan menambah data tambahan baku yang telah dibeli ke *database*.

#### 4.1.6 Diagram *activity*

Diagram *activity* digunakan untuk memodelkan aktivitas pengguna sistem berdasarkan *use case* skenario yang telah dibuat. Diagram *activity* yang dibuat adalah diagram *activity* mengelola pesanan, mengelola bahan baku, mengirim *push notification*, dan mengelola data bahan baku. *Activity diagram* digambarkan pada Gambar 4.4 hingga Gambar 4.7.

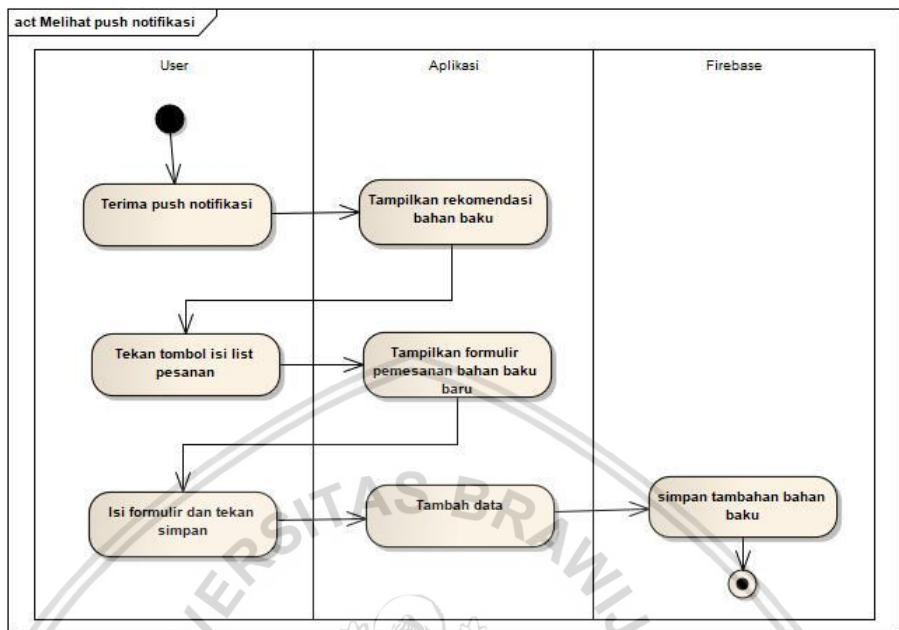


**Gambar 4.4 Activity Diagram Mengirim *Push Notification***

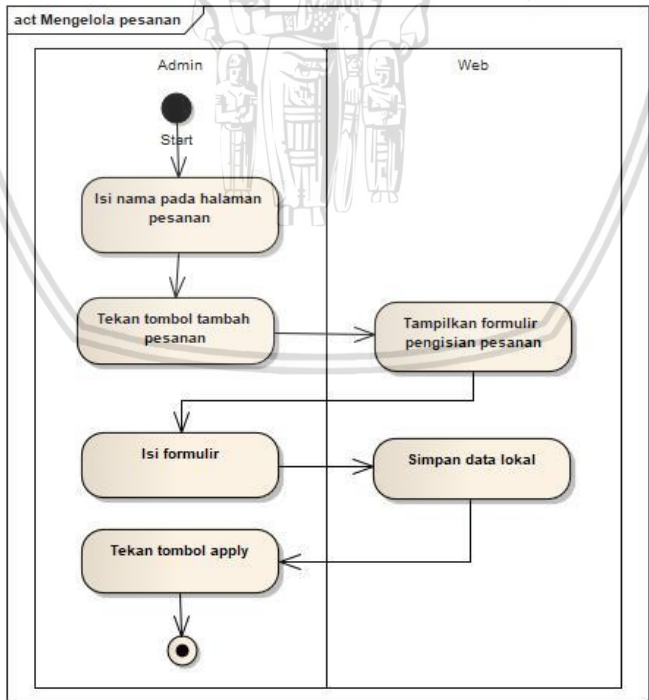
Gambar 4.4 menjelaskan mengenai aktivitas mengirim *push notification* yang dilakukan oleh administrator. Administrator menekan tombol kirim *push notification*, kemudian web akan menghitung rekomendasi bahan baku yang harus dibeli terlebih dahulu, dan disimpan ke *database*. Kemudian web akan memerintahkan Firebase untuk mengirim *push notification*.

Gambar 4.5 menjelaskan mengenai aktivitas dari mengelola data bahan baku. Setelah menerima *push notification*, aplikasi akan menampilkan rekomendasi bahan baku yang harus dibeli. Kemudian pemilik akan menekan nama bahan baku yang ada pada daftar. Aplikasi akan menampilkan formulir yang harus diisi.

Setelah pemilik kafe melengkapi formulir, aplikasi akan menambah jumlah bahan baku yang telah dibeli ke dalam database.



Gambar 4.5 Activity Diagram Mengelola Data Bahan Baku

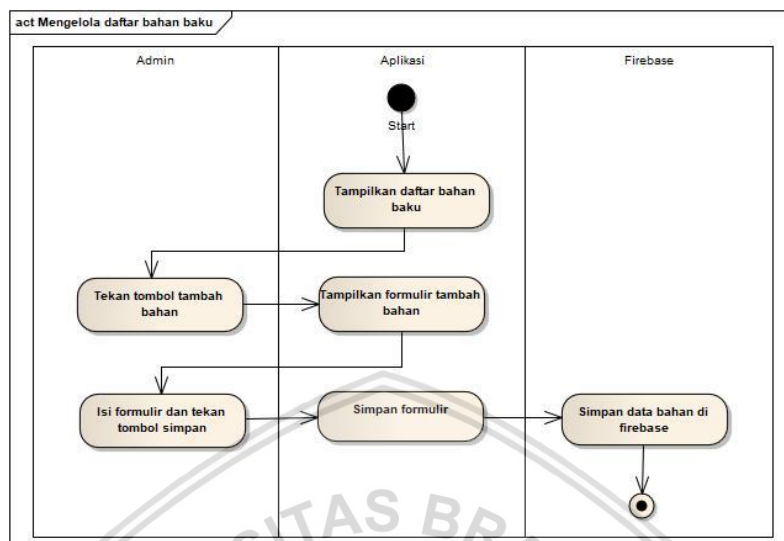


Gambar 4.6 Activity Diagram Mengelola Pesanan

Gambar 4.6 menggambarkan tentang *activity diagram* mengelola pesanan. Hal pertama yang harus dilakukan administrator adalah mengisi nama pada



halaman pesanan, kemudian menekan tombol tambah pesan, kemudian web akan menampilkan formulir pengisian pesanan. Setelah administrator mengisi formulir, maka web akan menyimpannya pada lokal data.



**Gambar 4.7 Activity Diagram Mengelola Daftar Bahan Baku**

Gambar 4.7 merupakan diagram *activity* dari mengelola daftar bahan baku. Hal pertama yang dilakukan administrator setelah berada pada halaman bahan baku adalah menekan tombol tambah bahan, kemudian mengisi formulir yang ditampilkan oleh web, kemudian web akan menyimpan data di *database*.

## 4.2 Perancangan

Dalam perancangan sistem dibahas mengenai perancangan arsitektural, diagram *sequence*, diagram *class*, desain antarmuka, perancangan algoritma, dan perancangan pengujian.

### 4.2.1 Perancangan arsitektural

Aplikasi pengelolaan bahan baku kafe menggunakan metode AHP dan *push notification* dibuat menggunakan teknologi *mobile* dengan memanfaatkan *push notification* sebagai fokus utama. *Push notification* dilakukan oleh Firebase Cloud Messaging atas permintaan dari aplikasi server.



**Gambar 4.8 Perancangan Arsitektur**

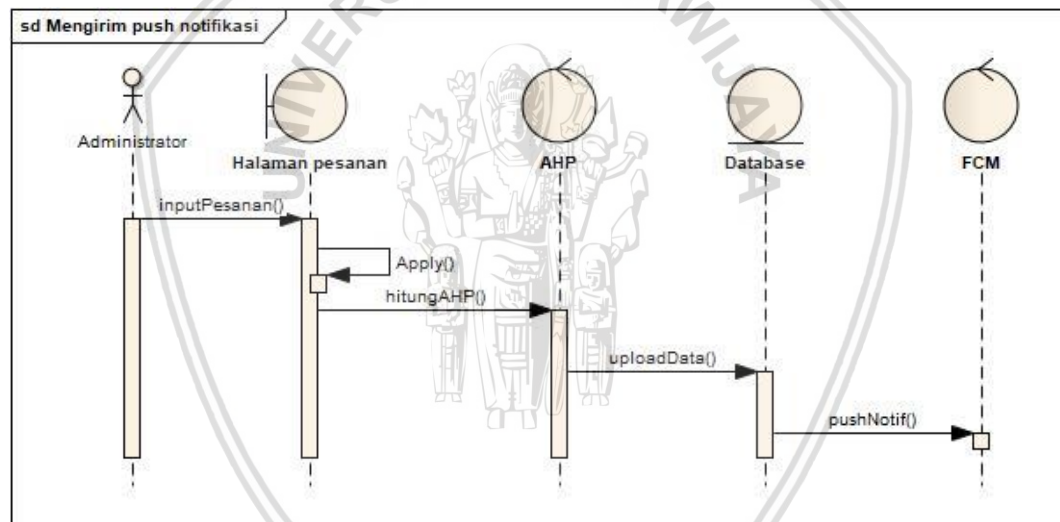
Gambar 4.8 adalah perancangan arsitektur pada aplikasi ini. Aplikasi *mobile* digunakan oleh pemilik kafe, sedangkan aplikasi *web-client* digunakan oleh administrator. Aplikasi *web-client* dibuat dengan teknologi web, menggunakan

*Vue.js*, sebuah javascript *framework*. Administrator dapat menekan sebuah tombol pada aplikasi *web-client* untuk melakukan perhitungan dengan menggunakan metode AHP sehingga aplikasi *web-client* memiliki hasil rekomendasi bahan baku yang harus dibeli dan kemudian dikirimkan kepada pemilik kafe dengan menggunakan *push notification*, melalui *Firestore*. Pemilik kafe pun dapat melakukan *update* data bahan baku yang ada pada *database*.

#### 4.2.2 Diagram *sequence*

Diagram *sequence* dibuat untuk memodelkan interaksi antara user dan objek pada suatu sistem dengan urutan waktu disertai langkah-langkah yang dilakukan untuk melakukan sebuah proses. Diagram *sequence* untuk aplikasi pengelolaan bahan baku kafe digambarkan dalam Gambar 4.9.

Gambar 4.9 menjelaskan mengenai proses mengirim *push notification* Administrator melakukan inputPesanan pada halaman pesan, kemudian halaman pesan melakukan *apply*, sehingga sistem akan menghitung AHP. Setelah dihitung, kemudian data di-*upload* ke *database*, kemudian data di *database* dikirim menggunakan *push notification*.

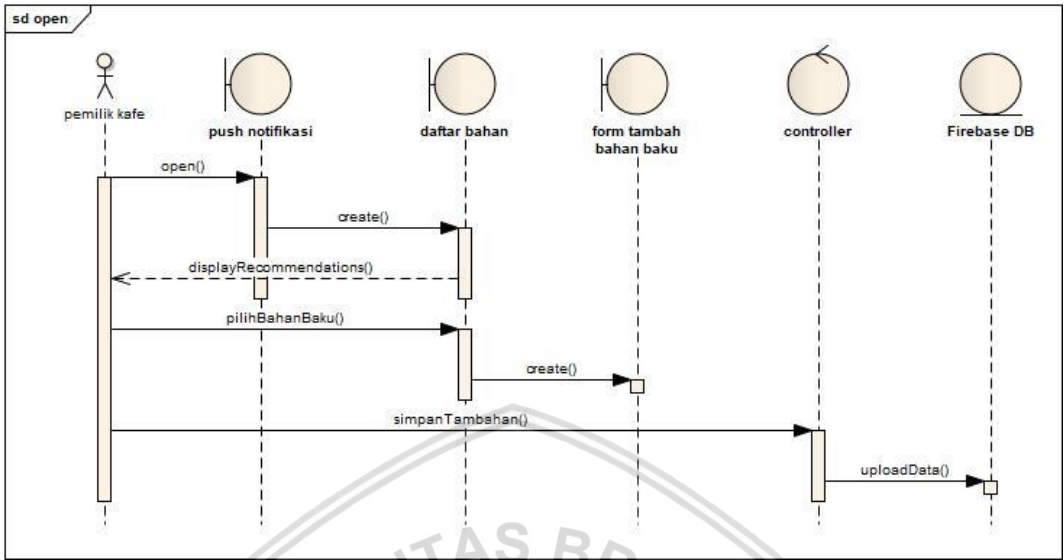


**Gambar 4.9 Diagram *Sequence* Mengirim *Push Notification***

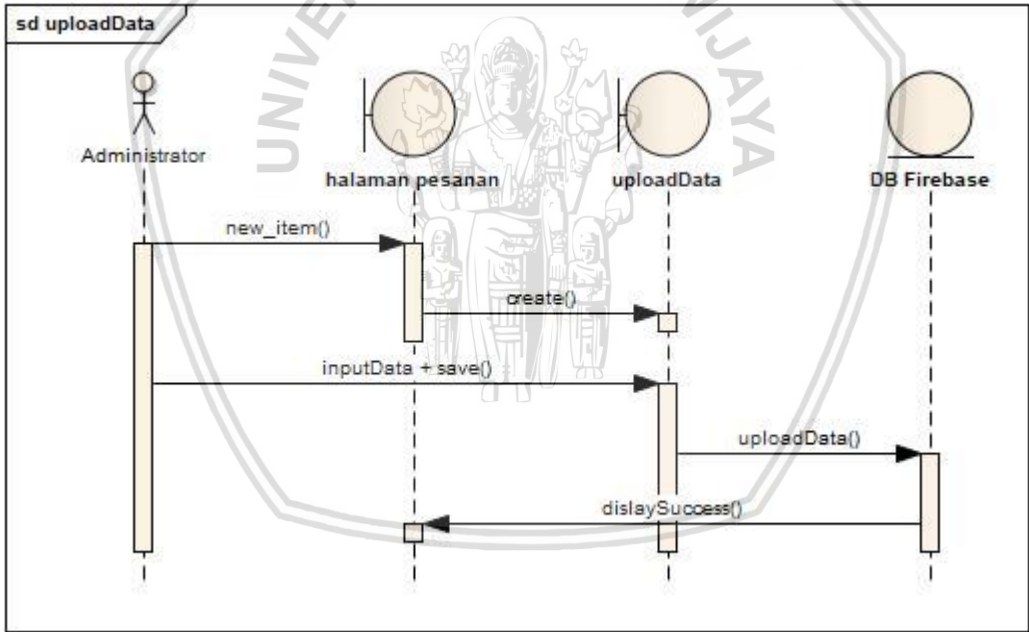
Gambar 4.10 merupakan diagram *sequence* dari pengelolaan bahan baku. Hal yang pertama dilakukan pemilik kafe adalah membuka *push notification* dengan fungsi *open()*, kemudian sistem melakukan *create()* daftar bahan, dan menampilkan rekomendasi dengan *displayRecomendations*. Kemudian pemilik kafe melakukan pilih bahan baku di halaman daftar bahan, dan membuat formulir tambah bahan baku. Kemudian pemilik menyimpan data tambahan dan *controller* mengunggah data ke *Firestore database*.

Gambar 4.11 merupakan diagram *sequence* mengelola pesanan. Administrator memasukkan pesanna baru dengan *new\_item()*, sehingga halaman pesan memanggil formulir tambah pesanan. Kemudian administrator

melakukan *inputData* dan *save()*, lalu melakukan *upload* data ke *Firestore database*.



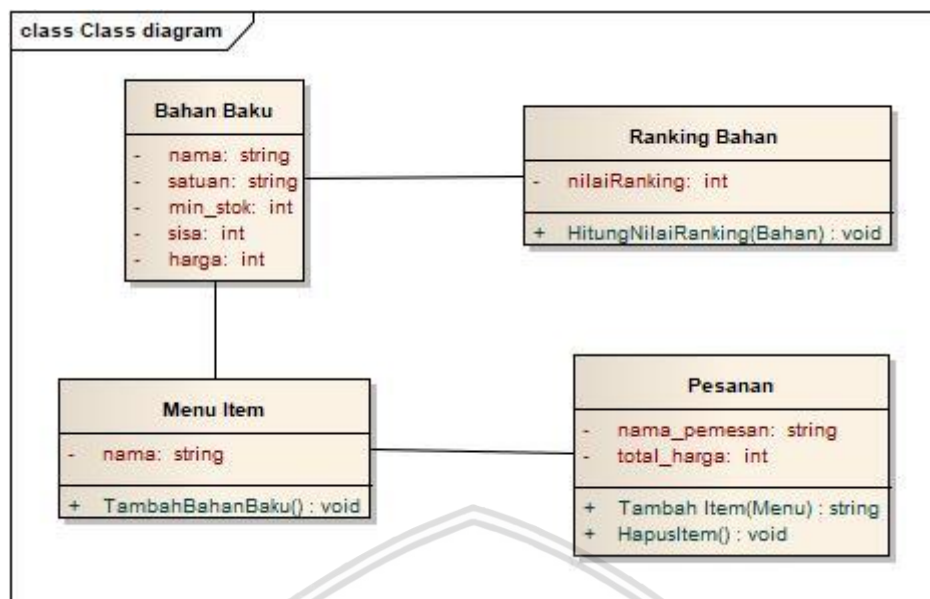
Gambar 4.10 Diagram Sequence Mengelola Bahan Baku



Gambar 4.11 Diagram Sequence Mengelola Pesanan

4.2.3 Diagram class

Diagram *class* dibuat untuk memodelkan struktur sistem dan relasi yang ada di dalamnya, yang berkaitan erat dengan *sequence* diagram. *Class* dan fungsi yang ada pada *sequence* diagram dirangkum sehingga menjadi *class* diagram pada Gambar 4.12. Terdapat empat class diagram, yaitu bahan baku, ranking bahan, menu item, dan pesanan.



**Gambar 4.12 Diagram Class**

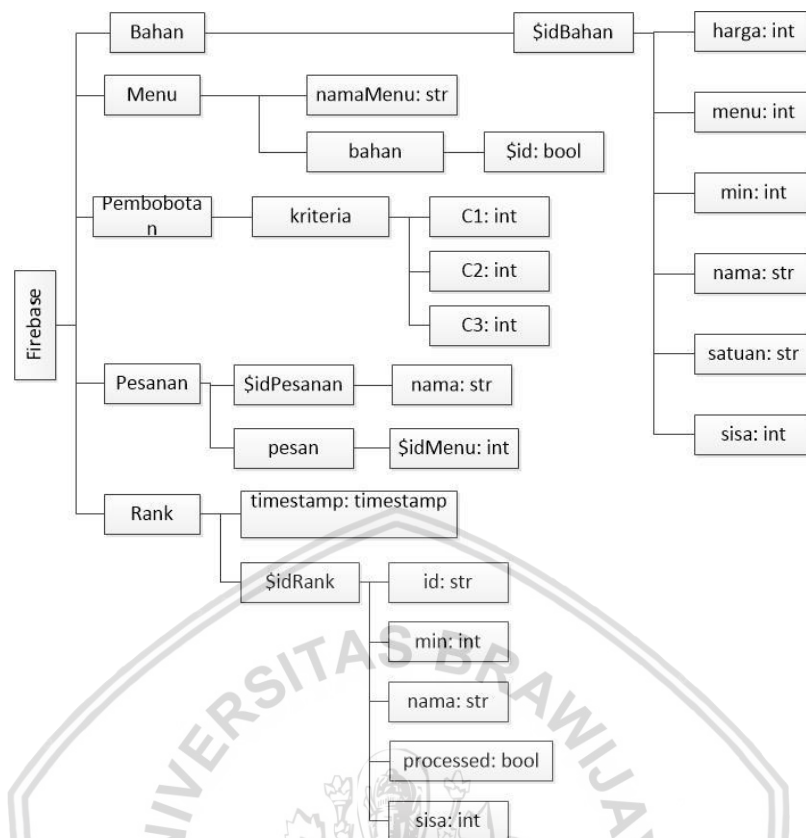
Gambar 4.12 merupakan perancangan diagram *class* sistem. Terdapat empat *class*, yaitu ranking bahan, bahan baku, menu item, dan pesanan. Bahan baku memiliki lima atribut, yaitu nama, satuan, min\_sisa, dan harga. Sedangkan ranking bahan memiliki atribut utama yaitu nilaiRanking berupa *integer*, dan memiliki atribut HitungnilaiRanking. Data yang diolah oleh *class* ranking bahan adalah data bahan.

#### 4.2.4 Perancangan basis data

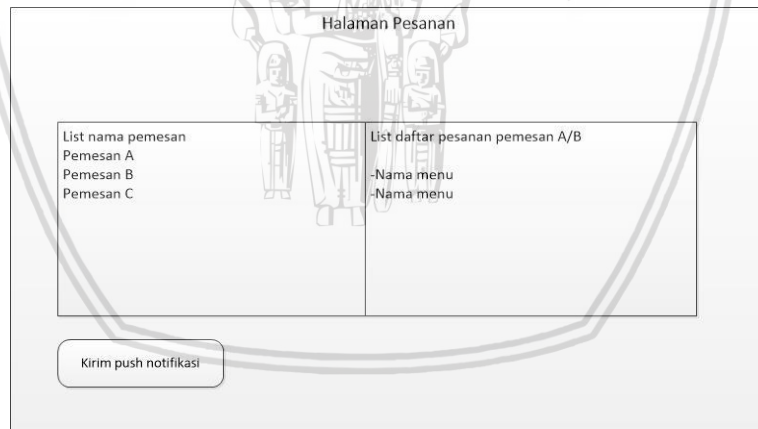
Basis data yang digunakan memiliki format JSON, dengan struktur yang berbeda dengan SQL. Perancangan basis data dapat digambarkan pada Gambar 4.13. Perancangan basis data dibuat berdasarkan *class* diagram yang telah dibuat sebelumnya. Diagram ini terdiri dari lima *node* utama, yaitu bahan, menu, pembobotan, pesanan, dan *rank*. Node pembobotan ditambahkan pada *database* untuk memudahkan bila ada fungsi baru untuk pergantian pembobotan kriteria AHP ndan membutuhkan pergantian data. *Node* bahan didapatkan dari *class diagram* bahan, yang memiliki lima atribut, namun ditambahkan *node* jumlah menu yang menggunakan bahan tersebut.

#### 4.2.5 Desain antarmuka

Desain antarmuka adalah desain sederhana yang akan menjadi acuan dalam pembuatan antarmuka aplikasi. Terdiri dari dua desain antarmuka utama, yaitu desain antarmuka untuk halaman pesanan yang ada pada *web-client* dan desain antarmuka halaman notifikasi yang ada pada aplikasi *mobile*.



**Gambar 4.13 Perancangan Database**



**Gambar 4.14 Desain Antarmuka Kirim Push Notification**

Gambar 4.14 merupakan desain antarmuka yang akan digunakan administrator untuk melakukan perhitungan dan mengirim *push notification* kepada pemilik kafe. Administrator dapat mengetahui nama pemesan dan nama menu yang dipesan oleh pengunjung kafe. Selain itu, administrator dapat mengirim *push notification* kepada pemilik dengan menekan tombol kirim *push notification*.

Gambar 4.15 merupakan desain antarmuka dari *push notification* yang diterima oleh pemilik kafe. Pemilik kafe mendapatkan *push notification* berupa list barang yang hampir habis dan harus dibeli, dan tombol masukkan pesanan.



Pemilik kafe dapat memasukkan pesanan setelah memesan maupun membeli bahan yang ada dalam daftar, sehingga administrator dapat melakukan validasi setelah barang tersebut sampai di kafe.

**Notifikasi**

-Tanggal/waktu

Bahan A

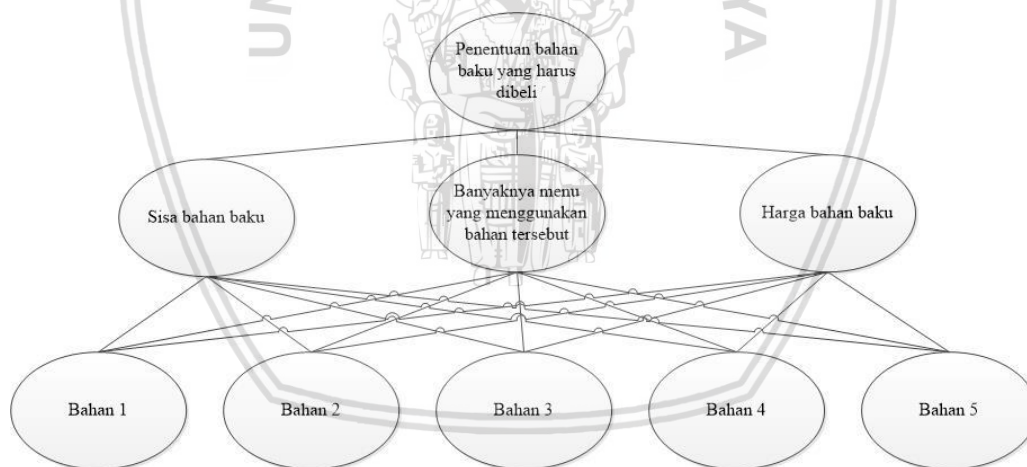
Bahan B

Bahan C

**Gambar 4.15** Desain antarmuka halaman notifikasi

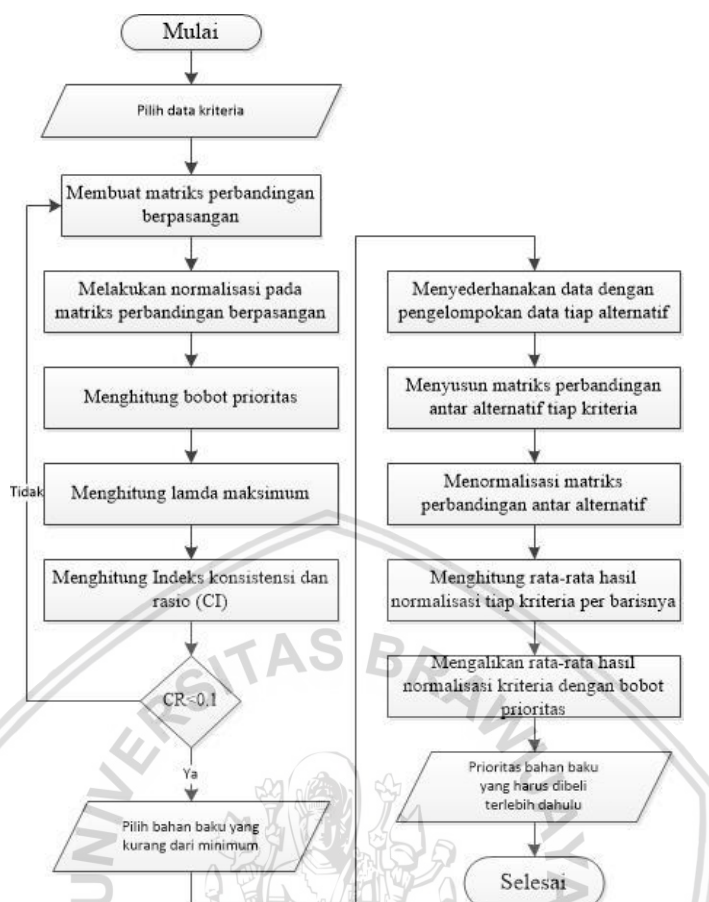
#### 4.2.6 Perancangan Algoritma AHP

Perancangan algoritma berisi mengenai cara kerja dan contoh penerapan metode AHP pada sistem ini. Algoritma AHP ini akan diterapkan pada *web-client*. Pohon hirarki metode AHP ditunjukkan pada Gambar 4.16.



**Gambar 4.16** Pohon Hirarki Metode AHP

Gambar 4.16 merupakan pohon hirarki dari penggunaan metode AHP pada sistem. Metode AHP digunakan untuk menentukan prioritas bahan baku yang harus segera dibeli, berdasarkan tiga kriteria, yaitu sisa stok bahan baku, banyaknya menu yang menggunakan bahan tersebut, dan harga bahan baku.



**Gambar 4.17 Diagram Alir Metode Ahp**

Gambar 4.17 merupakan diagram alir metode AHP, dimulai dari memilih data kriteria yang ada. Data kriteria didapatkan dari *database* yang telah ditentukan oleh pemilik kafe. Data kriteria kemudian dibuat menjadi matriks perbandingan berpasangan.

**Tabel 4. 1 Kriteria yang digunakan**

C1	Jumlah sisa bahan
C2	Banyaknya menu yang menggunakan bahan tersebut
C3	Harga bahan baku

Langkah kedua setelah ditentukannya kriteria adalah membuat matriks perbandingan. Nilai yang ada pada matriks perbandingan didapatkan berdasarkan pendapat pribadi pemilik kafe. Nilai matriks perbandingan dijelaskan pada Tabel 4. 2.

Tabel 4. 2 merupakan matriks perbandingan kriteria, dibuat berdasarkan Tabel 4.1. C1 sama pentingnya dengan C1, sehingga bernilai 1. C1 lebih penting bila dibandingkan dengan C2, sehingga bernilai lima. C1 jelas lebih penting bila dibandingkan dengan C3, sehingga bernilai tujuh. Sedangkan C2

sedikit lebih penting bila dibandingkan dengan C3, sehingga bernilai 3. Sedangkan C2 bernilai 1/5 bila dibandingkan dengan C1, karena nilai tersebut berkebalikan dengan nilai C1 bila dibandingkan dengan C2 adalah lima. Begitu pula dengan C3 bila dibandingkan dengan C1. Nilai tersebut merupakan kebalikan dari nilai perbandingan C1 dengan C3, yaitu 7, sehingga nilai C3 bila dibandingkan dengan C1 adalah 1/7. Hal selanjutnya yang dilakukan adalah melakukan normalisasi pada matriks perbandingan kriteria.

**Tabel 4. 2 Matriks perbandingan**

Kriteria	C1	C2	C3
C1	1	5	7
C2	0.2	1	3
C3	0.14	0.33	1

$$A = \begin{bmatrix} 1 & 5 & 7 \\ 0.2 & 1 & 3 \\ 0.14285714 & 0.333333 & 1 \end{bmatrix}$$

Matriks A didapatkan dari tabel perbandingan kriteria, yang kemudian akan dinormalisasikan, seperti pada rumus (2.1). Contoh normalisasi pada baris pertama kolom pertama:

$$\text{Normalisasi } A = \frac{1}{1 + 0.2 + 0.14285714} = 0.74468085$$

Perhitungan normalisasi dilakukan dengan semua nilai yang ada pada matriks, sehingga didapatkan nilai yang ada pada Tabel 4. 3.

**Tabel 4. 3 Hasil normalisasi matriks perbandingan kriteria**

Kriteria	C1	C2	C3
<b>C1</b>	0.74468085	0.789474	0.636364
<b>C2</b>	0.14893617	0.157895	0.272727
<b>C3</b>	0.10638298	0.052632	0.090909

Tabel 4. 3 merupakan hasil dari normalisasi yang telah dilakukan. Pada perbandingan C1 dengan C1, nilai yang semula 1 pada Tabel 4. 2 berubah menjadi 0.75 pada Tabel 4. 3 karena dilakukan normalisasi. Langkah selanjutnya adalah menghitung bobot kriteria. Bobot kriteria diperoleh dari perhitungan jumlah dari baris matriks normalisasi A dibagi dengan banyaknya kriteria, seperti pada rumus (2.2).

$$\text{Bobot Kriteria} = \frac{0.74468085 + 0.789474 + 0.636364}{3} = 0.723506$$

Perhitungan dilakukan pada seluruh baris yang ada pada Tabel 4. 3, sehingga matriks bobot kriteria adalah:

$$\text{Bobot kriteria} = \begin{bmatrix} 0.723506 \\ 0.193186 \\ 0.083308 \end{bmatrix}$$

Setelah ditemukan matriks bobot kriteria, hal yang dilakukan selanjutnya adalah menghitung  $\lambda_{max}$  maksimum, diawali dengan mengalikan matriks perbandingan dengan bobot kriteria. Berikut contoh perhitungan vektor konsistensi, baris sisa bahan baku. Perhitungan vektor konsistensi berdasarkan pada rumus (2.3) dan (2.4).

$$\begin{aligned} \text{Eigen kriteria bahan baku} &= \frac{(1 \times 0.723506) + (5 \times 0.193186) + (7 \times 0.083308)}{0.723506} \\ &= 3.14108156 \end{aligned}$$

Selanjutnya, mencari  $\lambda_{max}$  maksimum ( $\lambda_{max}$ ) didapatkan dengan mencari rata-rata dari nilai eigen seluruh kriteria.

$$\lambda_{max} = \frac{3.14108156 + 3.04271913 + 3.01365532}{3} = 3.06581867$$

$\lambda_{max}$  digunakan untuk menghitung nilai CI. Nilai CI didapatkan dengan menghitung  $\lambda_{max}$  dikurang jumlah banyaknya kriteria lalu dibagi jumlah kriteria yang dikurangi satu, seperti pada rumus (2.5).

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{3.06581867 - 3}{3 - 1} = 0.03290934$$

Nilai CI digunakan untuk mencari nilai *consistency ratio* yang berguna untuk mengetahui apakah matriks perbandingan yang telah dibuat konsisten atau tidak. *Consistency ratio* didapat dari CI dibagi dengan nilai RC, seperti pada rumus (2.6). Nilai RI sendiri didapatkan dari tabel RC (*Random Consistency*), dengan tiga kriteria, maka didapat nilai 0.58.

$$CR = \frac{CI}{RI} = \frac{0.0329034}{0.58} = 0.05674023$$

Nilai dari perhitungan CR adalah 0.05674023, sehingga matriks perbandingan tersebut dianggap konsisten karena  $CR < 0.1$ . Jika sudah konsisten, kemudian dilanjutkan dengan mencari nilai hasil akhir prioritas kriteria. Bahan baku yang memiliki sisa kurang dari batas minimum dijelaskan pada Tabel 4.7. Sisa bahan baku telah dikonversi menjadi gram, untuk menyetarakan satuan unit.

Tabel 4.7 berisi jumlah bahan baku yang tersisa pada kolom C1, jumlah menu yang menggunakan bahan baku tersebut pada kolom C2, serta harga bahan

baku pada terakhir pembelian pada kolom C3. Daftar bahan baku pada Tabel 4.7 dikelompokkan menjadi 9 tingkatan, digambarkan pada Tabel 4.8 hingga Tabel 4.10 berdasarkan kriterianya masing-masing.

**Tabel 4.7 Daftar Bahan Baku Yang Kurang Dari Batas Minimum**

Bahan baku	C1	C2	C3
sirup strawberry	400 gr	4	50000
es batu	1500 gr	14	50000
strawberry	1000 gr	3	39000
whipped cream	1800gr	7	76500
diamond full cream	9000 gr	4	15000

**Tabel 4.8 Pengelompokan Data Kriteria Sisa**

Data yang ada	Nilai
0-1333	9
1334-2667	8
2668-4001	7
4002-5335	6
5336-6669	5
6670-8003	4
8004-9337	3
9338-10671	2
10672-12000	1

Tabel 4.8 berisi pengelompokan data yang ada pada kriteria sisa bahan baku. Data terkecil yaitu 0 hingga 1.333 bernilai 9, yaitu nilai tertinggi yang berarti data tersebut memiliki prioritas tertinggi dibandingkan data yang lain pada kriteria sisa, sedangkan nilai 8 memiliki prioritas yang kurang dari nilai 9, tetapi memiliki prioritas lebih tinggi dibandingkan nilai yang lain.

Tabel 4.9 berisi pengelompokan data pada kriteria banyaknya menu yang menggunakan bahan tersebut. Nilai 9 didapatkan oleh data antara 33 hingga 36, sehingga data tersebut merupakan data yang harus diprioritaskan dalam kategori menu.

**Tabel 4.9 Pengelompokan Data Kriteria Menu Yang Menggunakan**

Data yang ada	Nilai
33-36	9
29-32	8
25-28	7
21-24	6
17-20	5
13-26	4
9-12	3
5-8	2
1-4	1

**Tabel 4.10 Pengelompokan Data Kriteria Harga**

Data yang ada	Nilai
0-44444	9
44445-88889	8
88890-133334	7
133335-177779	6
177780-222224	5
222225-266669	4
266670-311114	3
311115-355559	2
355560-400004	1

Pengelompokan data pada Tabel 4.8 berisi data yang ada pada kriteria harga. Data 0 hingga 44.444 dikelompokkan dengan nilai 9 sehingga data tersebut lebih prioritas dibandingkan dengan data yang lain. Pengelompokan dilakukan berdasarkan Tabel 4.11 hingga Tabel 4.10 sehingga didapatkan hasil pengelompokan pada Tabel 4.11.

**Tabel 4.11 Hasil pengelompokan data alternatif**

Nama	Alternatif	Sisa	Menu	Harga
B1	sirup strawberry	9	1	8
B2	es batu	8	4	8
B3	Strawberry	9	1	9
B4	whipped cream	8	2	8
B5	diamond full cream	3	1	9



Langkah selanjutnya adalah membuat matriks perbandingan antar alternatif dari masing-masing kriteria, seperti pada Tabel 4.12. Nilai 1 pada kolom B1 dari B1 didapat dari membagi kolom sisa baris B1 pada Tabel 4.11. Kemudian nilai 1.125 pada baris B2 dan kolom B2 pada Tabel 4.12 didapatkan dari membagi nilai baris B1 kolom sisa dengan nilai dari baris B2 kolom sisa pada Tabel 4.11.

**Tabel 4.12 Matriks perbandingan antar alternatif pada kriteria sisa bahan**

	B1	B2	B3	B4	B5
B1	1	1.125	1	1.125	3
B2	0.88888889	1	0.888889	1	2.666667
B3	1	1.125	1	1.125	3
B4	0.88888889	1	0.888889	1	2.666667
B5	0.33333333	0.375	0.333333	0.375	1

**Tabel 4.13 Normalisasi Matriks Perbandingan Antar Alternatif Kriteria Sisa**

	B1	B2	B3	B4	B5
B1	0.243243243	0.243243243	0.243243243	0.243243243	0.243243243
B2	0.216216216	0.216216216	0.216216216	0.216216216	0.216216216
B3	0.243243243	0.243243243	0.243243243	0.243243243	0.243243243
B4	0.216216216	0.216216216	0.216216216	0.216216216	0.216216216
B5	0.081081081	0.081081081	0.081081081	0.081081081	0.081081081

Tabel 4.13 berisi nilai rata-rata hasil normalisasi berdasarkan nilai pada Tabel 4.12. Berikut contoh perhitungan rata-rata hasil normalisasi matriks perbandingan antar alternatif pada kriteria sisa bahan, baris pertama. Hasil normalisasi tersebut diletakkan pada baris pertama kolom C1 pada Tabel 4.13.

$$\begin{aligned}
 &\text{Rata - rata} \\
 &= \frac{0.243243243 + 0.243243243 + 0.243243243 + 0.243243243 + 0.243243243}{5} \\
 &= 0.24324343
 \end{aligned}$$

**Tabel 4.14 Rata-rata Hasil Normalisasi**

Bahan baku	C1	C2	C3
sirup <i>strawberry</i>	0.24324324	0.111111	0.190476
es batu	0.21621622	0.444444	0.190476
<i>Strawberry</i>	0.24324324	0.111111	0.214286
<i>whipped cream</i>	0.21621622	0.222222	0.190476
<i>diamond full cream</i>	0.08108108	0.111111	0.214286

Setelah didapatkan rata-rata hasil normalisasi, langkah selanjutnya adalah mengalikan rata-rata hasil normalisasi pada Tabel 4.14 dengan bobot prioritas. Contoh perhitungan perkalian rata-rata hasil normalisasi pada nilai baris sirup *strawberry* Tabel 4.14 dengan bobot prioritas dijabarkan pada perhitungan hasil akhir berikut.

$$\begin{aligned} \text{Hasil akhir} &= (0.243243243 \times 3.14108156) + (0.1111111 \times 3.04271913) \\ &\quad + (0.19047619 \times 3.01365532) = 0.21333 \end{aligned}$$

Tabel 4.15 berisi nilai hasil akhir yang didapatkan dari seluruh nilai rata-rata hasil normalisasi pada Tabel 4.14 lalu dikalikan dengan bobot prioritas. Nilai terbesar merupakan bahan baku yang harus diprioritaskan terlebih dahulu, yaitu es batu.

**Tabel 4.15 Hasil akhir**

Bahan baku	Hasil akhir
sirup strawberry	0.213321
es batu	0.258162
Strawberry	0.215305
whipped cream	0.215232
diamond full cream	0.097979

#### 4.2.7 Perancangan pengujian

Pengujian pada sistem ini menggunakan pengujian fungsional menggunakan *blackbox*. Contoh pengujian fungsional dapat dilihat pada Tabel 4.16.

**Tabel 4.16 Tabel Pengujian Fungsional (*Blackbox*)**

Nomor	Kebutuhan	Skenario	Hasil
SRS-01	Administrator mengelola pesanan	Halaman pesanan	Valid/Tidak Valid
		Menekan tombol tambah pesanan	Valid/Tidak Valid
		Pesanan masuk ke database	Valid/Tidak Valid

Pada Tabel 4.16 terdapat lima kolom, yaitu Nomor, kebutuhan, skenario, tampilan, dan hasil. No. SRS, kebutuhan, dan skenario diambil dari Kebutuhan Fungsional dan Skenario Use Case dari masing-masing kebutuhan. Skenario ini akan dicocokkan dengan tampilan yang telah dibuat, apakah skenario kebutuhan yang diinginkan cocok dengan implementasi sistem.

Pengujian kedua yang dilakukan adalah pengujian akurasi AHP. Pengujian akurasi AHP dilakukan untuk mengetahui tingkat akurasi hasil perhitungan sistem bila dibandingkan dengan pendapat pemilik kafe. Bila urutan perankingan sistem sesuai dengan pendapat pemilik kafe, maka bernilai valid. Bila urutan perankingan berbeda, maka bernilai tidak valid.



## BAB 5 IMPLEMENTASI

Bab ini membahas mengenai pembuatan aplikasi pengelolaan bahan baku berdasarkan perancangan yang telah dibuat. Dalam bab implementasi terdiri dari batasan implementasi, implementasi basis data, implementasi kode program, dan implementasi antarmuka aplikasi.

### 5.1 Batasan implementasi

Batasan dalam mengimplementasikan sistem pengelolaan bahan baku kafe adalah aplikasi *mobile* maupun *web-client* harus terkoneksi internet agar dapat melakukan pertukaran data.

### 5.2 Implementasi basis data

Basis data diimplementasikan berdasarkan perancangan yang telah dibuat. Basis data ini menggunakan Firebase Real-Time Database dengan format JSON, seperti pada Tabel 5.1. Terdapat lima *node* utama, yaitu bahan, menu, pembobotan, pesanan, dan *rank*. *Node* bahan berisi data bahan baku yang ada di kafe. *Node* menu berisi data bahan baku yang dibutuhkan untuk membuat menu tersebut, beserta jumlah banyaknya bahan baku yang digunakan. *Node* pembobotan berisi nilai kriteria yang digunakan untuk AHP. *Node* pesanan digunakan untuk menyimpan data pesanan pelanggan, dan di dalamnya terdapat nama pemesan dan id pesanan. *Node* terakhir ada *node rank*, di dalamnya terdapat data bahan baku berdasarkan perhitungan AHP pada *web client*.

**Tabel 5.1 Implementasi struktur basis data**

1	{
2	"bahan" : {
3	"id_bahan" : {
4	"harga" : %float,
5	"menu" : %float,
6	"min" : %float,
7	"nama" : %string,
8	"satuan" : %string,
9	"sisa" : %float
10	}
11	},
12	"menu" : {
13	"id_menu" : {
14	"bahan" : {
15	id_bahan : %float
16	},
17	"namaMenu" : \$string
18	},
19	},
20	"pembobotan" : {
21	"id_kriteria" : {
22	"C1" : %float,
23	"C2" : %float,
24	"C3" : %float
25	}
26	},
27	"pesanan" : {

```

28     "id_pesanan" : {
29         "nama" : %string,
30         "pesan" : {
31             "idmenu" : %float
32         }
33     },
34     "rank" : {
35         id_rank : {
36             "1" : {
37                 "id" : %string,
38                 "min" : %float,
39                 "nama" : %string,
40                 "processed" : %boolean,
41                 "sisas" : %float
42             }
43         },
44         "timestamp" : %timestamp
45     },
46 }
47 }

```

### 5.3 Implementasi kode program

Kode program diimplementasikan berdasarkan perancangan pada bab sebelumnya. Tabel 5.2 adalah potongan kode program untuk menghitung rekomendasi bahan baku menggunakan AHP. Tabel 5.2 berada pada aplikasi *web-client*, yang dipanggil bila terdapat proses *update* bahan baku di *database*. Hasil perhitungan dari Tabel 5.2 tidak ditampilkan pada aplikasi *web-client*, melainkan disimpan ke *database node* rank. Hasil perhitungan implementasi AHP pada sistem sudah sesuai dengan perhitungan manual yang dilakukan pada tahap perancangan.

**Tabel 5.2 Kode implementasi AHP**

```

1     let kriteria = [];
2
3     this.kriteriaObj.forEach( content => {
4         kriteria.push([ content.C1, content.C2, content.C3 ]);
5     });
6
7     let length = kriteria.length;
8     let jumlahVer = [];
9
10    for (let i = 0; i < length; i++) {
11        let nilai=0;
12        for (let j = 0; j < length; j++) {
13            nilai += kriteria[j][i];
14        }
15        jumlahVer[i] = nilai;
16    }
17
18    let norm = new Array(length);
19    for (let i = 0; i < length; i++) {
20        norm [i] = new Array(length);
21        for (let j = 0; j < length; j++) {
22            norm[i][j] = kriteria[i][j] / jumlahVer[j];
23        }
24    }
25
26    length = norm.length;

```

27	let jumlahHorizontal = new Array(length);
28	for(let i=0; i < length; i++) {
29	let plus = 0;
30	for(let j=0; j < length; j++) {
31	plus += norm[i][j];
32	}
33	jumlahHorizontal[i] = plus;
34	}
35	
36	let bobotPrioritas = new Array(length);
37	let bobot = 0;
38	for (let i=0; i < length; i++) {
39	bobot = jumlahHorizontal[i] / 3;
40	bobotPrioritas[i] = bobot;
41	}
42	
43	let eigenKriteria = new Array(length);
44	let eigen = [];
45	let bagi = 0;
46	for(let i = 0; i < length; i++){
47	let kali =0;
48	for(let j = 0; j < length; j++){
49	kali += (kriteria[i][j] * bobotPrioritas[j]);
50	eigen[i] = kali;
51	}
52	}
53	for(let i=0; i < length; i++){
54	bagi = eigen[i] / bobotPrioritas[i];
55	eigenKriteria[i] = bagi;
56	}
57	
58	let lamdaMax = 0;
59	eigen =0;
60	for(let i=0; i < length; i++) {
61	eigen += eigenKriteria[i];
62	}
63	lamdaMax = eigen/3;
64	
65	let ci = 0;
66	ci = (lamdaMax - 3)/(3-1);
67	let cr =0;
68	cr = ci/0.58;
69	
70	let lessThanMin = [];
71	let altIdName = [];
72	this.getLessThanMin().forEach(item => {
73	altIdName.push([item[0], item[1]]);
74	lessThanMin.push([item[2], item[3], item[4], item[5]]);
75	});
76	if (lessThanMin.length == 0) {return};
77	
78	
79	let alternatif = [];
80	this.getLessThanMin().forEach(item => {
81	alternatif.push([item[2], item[3], item[4]]);
82	});
83	
84	for(let i=0; i < alternatif.length; i++){
85	if(alternatif[i][0] > 0 && alternatif[i][0] < 1334){
86	alternatif[i][0] = 9;
87	}
88	else if(alternatif[i][0] > 1333 && alternatif[i][0] <
89	2668){
90	alternatif[i][0] = 8;
91	}



```

92     else if(alternatif[i][0] > 2667 && alternatif[i][0] <
93     4002){
94         alternatif[i][0] = 7;
95     }
96     else if(alternatif[i][0] > 4001 && alternatif[i][0] <
97     5336){
98         alternatif[i][0] = 6;
99     }
100    else if(alternatif[i][0] > 5335 && alternatif[i][0] <
101    6670){
102        alternatif[i][0] = 5;
103    }
104    else if(alternatif[i][0] > 6669 && alternatif[i][0] <
105    8004){
106        alternatif[i][0] = 4;
107    }
108    else if(alternatif[i][0] > 8003 && alternatif[i][0] <
109    9338){
110        alternatif[i][0] = 3;
111    }
112    else if(alternatif[i][0] > 9337 && alternatif[i][0] <
113    10672){
114        alternatif[i][0] = 2;
115    }
116    else{
117        alternatif[i][0] = 9;
118    }
119 }
120 for(let i=0; i < alternatif.length; i++){
121     if(alternatif[i][1] > 32 && alternatif[i][1] < 37){
122         alternatif[i][1] = 9;
123     }
124     else if(alternatif[i][1] > 28 && alternatif[i][1] <
125     33){
126         alternatif[i][1] = 8;
127     }
128     else if(alternatif[i][1] > 24 && alternatif[i][1] <
129     29){
130         alternatif[i][1] = 7;
131     }
132     else if(alternatif[i][1] > 20 && alternatif[i][1] <
133     25){
134         alternatif[i][1] = 6;
135     }
136     else if(alternatif[i][1] > 16 && alternatif[i][1] <
137     21){
138
139         alternatif[i][1] = 5;
140     }
141     else if(alternatif[i][1] > 12 && alternatif[i][1] <
142     17){
143         alternatif[i][1] = 4;
144     }
145     else if(alternatif[i][1] > 8 && alternatif[i][1] < 13){
146         alternatif[i][1] = 3;
147     }
148     else if(alternatif[i][1] > 4 && alternatif[i][1] < 9){
149         alternatif[i][1] = 2;
150     }
151     else if(alternatif[i][1] < 5) {
152         alternatif[i][1] = 1;
153     }
154 }
155 for(let i=0; i < alternatif.length; i++){
156     if(alternatif[i][2] > 0 && alternatif[i][2] < 44443){

```

```

157         alternatif[i][2] = 9;
158     }
159     else if(alternatif[i][2] > 44444 && alternatif[i][2] <
160 88888){
161         alternatif[i][2] = 8;
162     }
163     else if(alternatif[i][2] > 88889 && alternatif[i][2] <
164 133335){
165         alternatif[i][2] = 7;
166     }
167     else if(alternatif[i][2] > 133334 && alternatif[i][2] <
168 177780){
169         alternatif[i][2] = 6;
170     }
171     else if(alternatif[i][2] > 177779 && alternatif[i][2] <
172 222225){
173         alternatif[i][2] = 5;
174     }
175     else if(alternatif[i][2] > 222224 && alternatif[i][2] <
176 266670){
177         alternatif[i][2] = 4;
178     }
179     else if(alternatif[i][2] > 266669 && alternatif[i][2] <
180 311115){
181         alternatif[i][2] = 3;
182     }
183     else if(alternatif[i][2] > 311115 && alternatif[i][2] <
184 355560){
185         alternatif[i][2] = 2;
186     }
187     else{
188         alternatif[i][2] = 1;
189     }
190 }
191
192 let eigenKriteriaSisa = [];
193
194 for (let i = 0; i < alternatif.length; i++) {
195     let rowEigen = [];
196     for (let j = 0; j < alternatif.length; j++){
197         if (alternatif.hasOwnProperty(j) &&
198 alternatif.hasOwnProperty(i)) {
199             let bagi = alternatif[i][0] / alternatif[j][0];
200             rowEigen[j] = bagi;
201         }
202     }
203     eigenKriteriaSisa[i] = rowEigen;
204 }
205 let jumlahVerAS = new Array(alternatif.length);
206 for (let i = 0; i < alternatif.length; i++) {
207     let nilai=0;
208     for (let j = 0; j < alternatif.length; j++) {
209         nilai += eigenKriteriaSisa[j][i];
210     }
211     jumlahVerAS[i] = nilai;
212 }
213
214 let normAltSisa = new Array(alternatif.length);
215 for (let i = 0; i < alternatif.length; i++) {
216     normAltSisa [i] = new Array(alternatif.length);
217     for (let j = 0; j < alternatif.length; j++) {
218         normAltSisa[i][j] = eigenKriteriaSisa[i][j] /
219 jumlahVerAS[j];
220     }
221 }

```

```

222
223     let eigenKriteriaMenu = []
224     for (let i = 0; i < alternatif.length; i++) {
225         let rowEigen = [];
226         for (let j = 0; j < alternatif.length; j++){
227             if (alternatif.hasOwnProperty(j)                &&
228 alternatif.hasOwnProperty(i)) {
229                 let bagi = alternatif[i][1] / alternatif[j][1];
230                 // console.log(alternatif[i][0]            +"/"+
231 alternatif[j][0] + " = " + bagi );
232                 rowEigen[j] = bagi;
233             }
234         }
235         eigenKriteriaMenu[i] = rowEigen;
236     }
237
238     let jumlahVerAM = new Array(alternatif.length);
239     for (let i = 0; i < alternatif.length; i++) {
240         let nilai=0;
241         for (let j = 0; j < alternatif.length; j++) {
242             nilai += eigenKriteriaMenu[j][i];
243         }
244         jumlahVerAM[i] = nilai;
245     }
246
247     let normAltMenu = new Array(alternatif.length);
248     for (let i = 0; i < alternatif.length; i++) {
249         normAltMenu [i] = new Array(alternatif.length);
250         for (let j = 0; j < alternatif.length; j++) {
251             normAltMenu[i][j] = eigenKriteriaMenu[i][j] /
252 jumlahVerAM[j];
253         }
254     }
255
256     let eigenKriteriaHarga = [];
257     for (let i = 0; i < alternatif.length; i++) {
258         let rowEigen = [];
259         for (let j = 0; j < alternatif.length; j++){
260             if (alternatif.hasOwnProperty(j)                &&
261 alternatif.hasOwnProperty(i)) {
262                 let bagi = alternatif[i][2] / alternatif[j][2];
263                 rowEigen[j] = bagi;
264             }
265         }
266         eigenKriteriaHarga[i] = rowEigen;
267     }
268
269     let jumlahVerAH = new Array(alternatif.length);
270     for (let i = 0; i < alternatif.length; i++) {
271         let nilai=0;
272         for (let j = 0; j < alternatif.length; j++) {
273             nilai += eigenKriteriaHarga[j][i];
274         }
275         jumlahVerAH[i] = nilai;
276     }
277
278     let normAltHarga = new Array(alternatif.length);
279     for (let i = 0; i < alternatif.length; i++) {
280         normAltHarga [i] = new Array(alternatif.length);
281         for (let j = 0; j < alternatif.length; j++) {
282             normAltHarga[i][j] = eigenKriteriaHarga[i][j] /
283 jumlahVerAH[j];
284         }
285     }
286

```

```

287 let jumlahHorSisa = new Array(normAltSisa.length);
288 for (let i = 0; i < alternatif.length; i++) {
289     let nilai=0;
290     for (let j = 0; j < alternatif.length; j++) {
291         nilai += normAltSisa[i][j];
292     }
293     jumlahHorSisa[i] = nilai/5;
294 }
295
296 let jumlahHorMenu = new Array(normAltMenu.length);
297 for (let i = 0; i < alternatif.length; i++) {
298     let nilai=0;
299     for (let j = 0; j < alternatif.length; j++) {
300         nilai += normAltMenu[i][j];
301     }
302     jumlahHorMenu[i] = nilai/5;
303 }
304
305 let jumlahHorHarga = new Array(normAltHarga.length);
306 for (let i = 0; i < alternatif.length; i++) {
307     let nilai=0;
308     for (let j = 0; j < alternatif.length; j++) {
309         nilai += normAltHarga[i][j];
310     }
311     jumlahHorHarga[i] = nilai/5;
312 }
313
314 eigenKriteria = [];
315 for(let i=0; i<normAltSisa.length; i++){
316     eigenKriteria[i] = new Array(kriteria.length);
317     for(let j=0; j < kriteria.length; j++){
318         eigenKriteria[i][0] = jumlahHorSisa[i];
319         eigenKriteria[i][1] = jumlahHorMenu[i];
320         eigenKriteria[i][2] = jumlahHorHarga[i];
321     }
322 }
323
324 let rank = new Array(alternatif.length);
325 for(let i = 0; i < alternatif.length; i++){
326     let kali =0;
327     for(let j = 0; j < kriteria.length; j++){
328         kali += (eigenKriteria[i][j] * bobotPrioritas[j]);
329         // console.log(eigenKriteria[i][j] + "*" +
330 bobotPrioritas[j] + " = " + kali );
331         rank[i] = kali;
332     }
333 }
334
335 let rankData = [];
336 for (let i=0; i < lessThanMin.length; i++){
337     rankData[i] = new Array(kriteria.length);
338     // for(j=0; j < kriteria.length; j++){
339     rankData[i][0] = altIdName[i][0]; // id
340     rankData[i][1] = altIdName[i][1]; // nama
341     rankData[i][2] = rank[i]; // nilai perhitungan ranking
342     rankData[i][3] = lessThanMin[i][0]; // sisa
343     rankData[i][4] = lessThanMin[i][3]; // minimum
344     // }
345 }
346
347 rankData.sort(function(a,b){
348     if (a[2] === b[2]){
349         return 0;
350     }
351     else{

```

352	return (a[2]>b[2]) ? -1 : 1;
353	}
354	})

Berikut merupakan penjelasan dari Tabel 5.2.

1. Baris 1-8 merupakan pengambilan kriteria dari *database* Firebase, agar dapat menjadi matriks pendapat individu.
2. Baris 10-16 merupakan proses penjumlahan vertikal pada matriks pendapat individu.
3. Baris 18-24 merupakan normalisasi matriks pendapat individu.
4. Baris 26-41 merupakan perhitungan bobot prioritas dengan cara menjumlah masing-masing kolom yang ada pada matriks pendapat individu lalu membaginya dengan tiga.
5. Baris 43-56 merupakan perhitungan vektor konsistensi.
6. Baris 58-63 merupakan perhitungan *lamda max*, yang didapat dengan cara menghitung rata-rata dari vektor konsistensi.
7. Baris 65-68 merupakan penulisan kode untuk perhitungan CI, menentukan apakah tabel kriteria telah konsisten.
8. Baris 70-77 merupakan pengambilan data alternatif dari *database*
9. Baris 79-190 merupakan pengelompokan data alternatif.
10. Baris 192-285 merupakan perhitungan matriks perbandingan antar alternatif.
11. Baris 287-322 merupakan perhitungan normalisasi matriks perbandingan antar alternatif.
12. Baris 324-354 merupakan perhitungan ranking rekomendasi.

Untuk pengiriman *push notification*, digunakan salah satu produk Firebase, yaitu Firebase Cloud Functions. Tabel 5.3 merupakan kode dari Firebase Cloud Functions, yang merupakan *trigger* agar Firebase Cloud Messaging untuk mengirim pesan kepada aplikasi *mobile* pemilik kafe.

**Tabel 5.3 Implementasi *push notification* menggunakan Firebase Cloud Functions**

1	var functions = require('firebase-functions');
2	var admin = require('firebase-admin');
3	admin.initializeApp();
4	
5	exports.updateStockNotif = functions.database.ref('/rank/{rankId}')
6	.onCreate((change, context) => {
7	var payload = {
8	notification: {
9	title: "Update Stock Bahan",
10	body: "Content update stock bahan"
11	}
12	};
13	var rankId = context.params.rankId;
14	return admin.messaging().sendToTopic("rank", payload);

15	});
----	-----

Berikut merupakan penjelasan dari Tabel 5.3.

1. Pada baris pertama hingga ketiga merupakan kode untuk mengimpor modul Firebase yang dibutuhkan, yaitu Firebase Functions dan Firebase Admin.
2. Baris 5 hingga 12 memiliki arti bila terdapat *update* data pada *database* rank yang berupa penambahan rankId, maka Firebase Cloud Messaging akan mengirim *push notification* dengan judul "Update stock bahan" dan berisi "Content update stock bahan".
3. Pada baris 13 hingga 15 berisi pengiriman topik *push notification* yang berisi data yang ada pada tabel rank.

**Tabel 5.4 Kode program melihat *push notification* dan *update* bahan baku**

1	Public class NotificationResultActivity extends AppCompatActivity {	
2	private DatabaseReference mDatabase;	
3	private RecyclerView mRecyclerView;	
4	private View mIndicator;	
5	private RankAdapter mAdapter;	
6	private ChildEventListener mDataListener;	
7		
8	@Override	
9	protected void onCreate(Bundle savedInstanceState) {	
10	super.onCreate(savedInstanceState);	
11	FirebaseAuth mAuth = FirebaseAuth.getInstance();	
12	UserHelper.checkLogin(mAuth, ()	->
13	UserHelper.runLogin(this));	
14		
15	setContentView(R.layout.activity_notification_result);	
16	mIndicator = findViewById(R.id.itemsIndicatorGroup);	
17	mRecyclerView = findViewById(R.id.itemsRecyclerView);	
18	mAdapter = new RankAdapter(this);	
19	mAdapter.setOnClickListener(this::runUpdateIngredient);	
20	mRecyclerView.setLayoutManager(new LinearLayoutManager(this));	
21	mRecyclerView.setAdapter(mAdapter);	
22		
23	mDatabase	=
24	FirebaseDatabase.getInstance().getReference("rank");	
25		
26	@Override	
27	protected void onStart() {	
28	super.onStart();	
29	listenData();	
30	}	
31		
32	@Override	
33	protected void onStop() {	
34	super.onStop();	
35	mDatabase.removeEventListener(mDataListener);	
36	}	
37		
38	private void listenData() {	
39	Timber.d("Listen Data");	
40	mAdapter.clear();	
41	mIndicator.setVisibility(View.VISIBLE);	
42	mRecyclerView.setVisibility(View.INVISIBLE);	
43	mDataListener = new DataChildListener(	
44	(dataSnapshot, previousChildName) -> {	
45	Rank data	=



```

46 RankHelper.convertDataSnapshot(dataSnapshot);
47
48         mAdapter.add(data);
49
50         // Show list and hide loading indicator after
51 add data
52         mIndicator.setVisibility(View.GONE);
53         mRecycler.setVisibility(View.VISIBLE);
54         Timber.d("On Child Added, %s", data);
55     },
56     (dataSnapshot, previousChildName) -> {
57         Rank data =
58 RankHelper.convertDataSnapshot(dataSnapshot);
59         mAdapter.change(data);
60         Timber.d("On Child Changed, %s", data);
61     },
62     dataSnapshot -> {
63         Rank data =
64 RankHelper.convertDataSnapshot(dataSnapshot);
65         mAdapter.remove(data);
66         Timber.d("On Child Removed, %s", data);
67     },
68     (dataSnapshot, previousChildName) -> Timber.d("On
69 Child Moved"),
70     databaseError -> Timber.d("On Child Canceled")
71 );
72
73 mDatabase.orderByChild("timestamp").addChildEventListener(mDataListe
74 ner);
75 }
76
77 private void runUpdateIngredient(String id, String rankId,
78 String key) {
79     if (id == null || rankId == null) return;
80
81     Intent intent = new Intent(this,
82 UpdateIngredientActivity.class);
83     intent.putExtra("ITEM_ID", id);
84     intent.putExtra("RANK_ID", rankId);
85     intent.putExtra("KEY", key);
86     startActivity(intent);
87 }
88 }

```

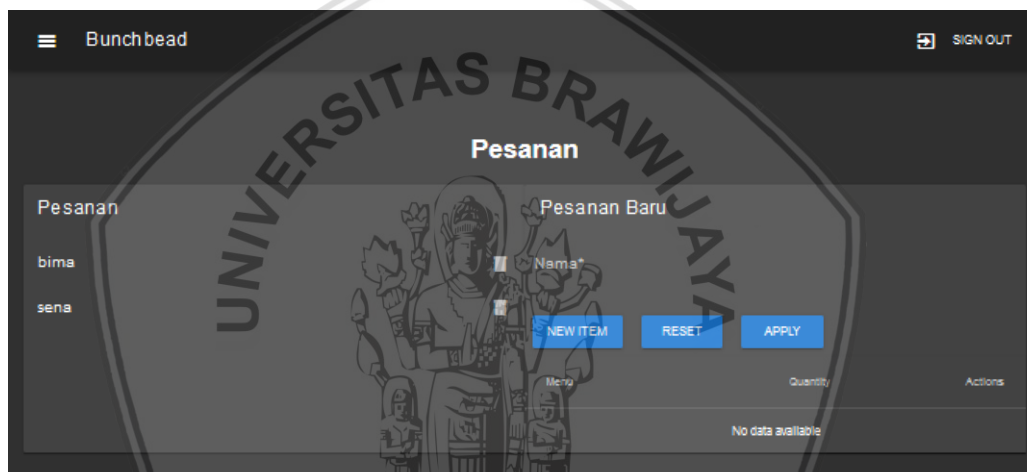
Tabel 5.4 menunjukkan potongan kode untuk proses melihat *push notification* dan melakukan *update* data bahan baku yang pada aplikasi *mobile* pemilik kafe. Kode program ini terletak pada *class NotificationResultActivity*.

13. Baris 1-21 merupakan inialisasi variabel.
14. Baris 23-24 merupakan pemanggilan *database*, data yang digunakan pada *node* rank.
15. Baris 26-30 aplikasi akan mendengarkan pertukaran data saat aplikasi dimulai.
16. Pada baris 32-36 aplikasi akan berhenti mendengarkan pertukaran data saat aplikasi berhenti.

17. Baris 38-75 merupakan fungsi *listenData* yang berisi pengambilan data dari *node* rank saat fungsi tersebut dijalankan, dan data tersebut diurutkan berdasarkan waktu data tersebut dimasukkan ke *database*.
18. Baris 77-79 merupakan fungsi untuk melakukan *update* bahan baku.
19. Baris 82-88 dilakukan saat pemilik kafe melakukan *update* bahan baku. Data yang diperlukan untuk melakukan *update* data bahan adalah ID bahan dan ID rank.

## A.2 Implementasi antarmuka

Implementasi antarmuka aplikasi pengelolaan bahan baku kafe dilakukan berdasarkan perancangan yang telah dibuat. *Web-client* dibuat dengan menggunakan platform Vue.js, sehingga tampilan dibuat dengan menggunakan HTML dan javascript.



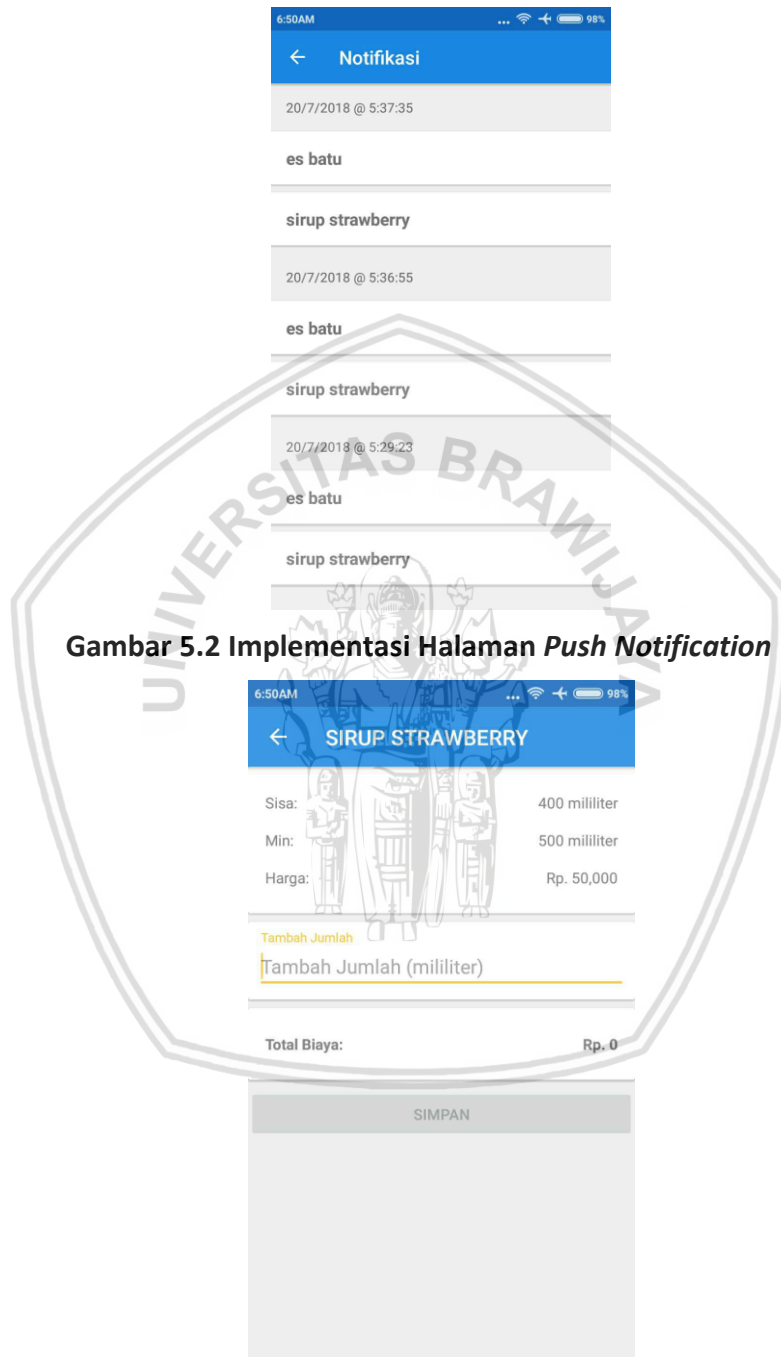
**Gambar 5.1 Implementasi Antarmuka Halaman Pesanan**

Gambar 5.1 merupakan implementasi dari halaman pesanan. Dalam halaman pesanan terdapat daftar nama pemesan di sebelah kiri, dan administrator dapat melakukan penambahan nama pemesan dan pesanan di sebelah kanan. Tombol *new item* digunakan untuk menambah pesanan baru, sedangkan tombol *apply* digunakan untuk memasukkan data ke *database*, menghitung rekomendasi menggunakan AHP, dan mengirim *push notification* sekaligus.

Gambar 5.2 dan Gambar 5.3 dibuat pada halaman aplikasi *mobile* pemilik kafe. Gambar 5.2 diimplementasikan pada *activity\_notification\_result.xml*, sedangkan Gambar 5.3 diimplementasikan pada *activity\_update\_ingredient.xml*.

Gambar 5.2 merupakan tampilan halaman daftar *push notification* yang ada pada perangkat *mobile*. Tampilan Daftar ini berisi rekomendasi bahan baku secara berurutan, baris pertama adalah bahan baku yang harus dibeli terlebih dahulu.

Gambar 5.3 merupakan tampilan halaman menambah jumlah bahan baku berdasarkan daftar *push notification* yang ada. Dalam halaman ini pemilik kafe dapat menambahkan bahan baku secara langsung ke dalam *database*.



Gambar 5.2 Implementasi Halaman *Push Notification*

Gambar 5.3 Implementasi Halaman Menambah Jumlah Bahan Baku

## BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis membahas mengenai pengujian serta analisis aplikasi pengelolaan bahan baku kafe yang telah diimplementasikan. Hasil pengujian kemudian dianalisis dan dijadikan pengukuran apakah target penelitian tercapai. Pengujian terdiri dari dua metode, yaitu pengujian validasi dan pengujian.

### 6.1 Pengujian

Pengujian yang dilakukan terhadap aplikasi pengelolaan bahan baku terdiri dari dua pengujian, yaitu pengujian validasi dan pengujian akurasi. Pengujian validasi menggunakan teknik *blackbox*, sedangkan untuk pengujian akurasi berdasarkan perbandingan aplikasi pengelolaan bahan baku dengan pendapat pemilik kafe.

### 6.2 Pengujian validasi

Pengujian validasi ditujukan untuk mengetahui apakah aplikasi yang dibuat sesuai dengan kebutuhan yang telah dirumuskan. Pengujian validasi menggunakan *blackbox*.

**Tabel 6.1 Kasus Uji Mengelola Pesanan**

ID Kasus uji	TC-01
Nama kasus uji	Mengelola pesanan
ID Use case	SRS-01
Tujuan pengujian	Membuktikan bahwa <i>web-client</i> dapat mengelola daftar pesanan
Prosedur pengujian	1. Menekan tombol <i>new item</i> 2. Memasukkan pesanan ke formulir 3. Menekan tombol <i>save</i>
Hasil yang diharapkan	<i>Web-client</i> dapat memasukkan data pesanan ke <i>database</i>

**Tabel 6.2 Kasus Uji Mengelola Bahan Baku**

ID Kasus uji	TC-02
Nama kasus uji	Mengelola bahan baku
ID Use case	SRS-02
Tujuan pengujian	Membuktikan bahwa <i>web-client</i> dapat menambah data bahan baku baru

Tabel 6.3 Kasus Uji Mengelola Bahan Baku (lanjutan)

Prosedur pengujian	1. Menekan tombol tambah bahan 2. Mengisi formulir Menekan tombol simpan
Hasil yang diharapkan	<i>Web-client</i> dapat memasukkan data bahan baku ke <i>database</i>

Tabel 6.4 Kasus Uji Mengirim *Push Notification*

ID Kasus uji	TC-03
Nama kasus uji	Mengirim <i>push notification</i>
ID Use case	SRS-03
Tujuan pengujian	Membuktikan bahwa aplikasi <i>web client</i> memiliki fitur pengiriman <i>push notification</i> dan bisa berjalan dengan baik
Prosedur pengujian	1. Memasukkan pesanan 2. Menekan tombol <i>apply</i>
Hasil yang diharapkan	Sistem dapat mengirim <i>push notification</i> ke aplikasi <i>mobile</i> yang berisi hasil rekomendasi perhitungan AHP nila ada bahan baku yang kurang dari minimum

Tabel 6.5 Kasus Uji Melihat *Push Notification*

ID Kasus uji	TC-04
Nama kasus uji	Melihat <i>push notification</i>
ID Use case	SRS-04
Tujuan pengujian	Membuktikan bahwa aplikasi <i>mobile</i> dapat menerima dan menampilkan <i>push notification</i> yang dikirimkan oleh <i>web client</i>
Prosedur pengujian	1. Mendapatkan <i>push notification</i> 2. Membuka halaman <i>push notification</i> 3. Mengecek daftar rekomendasi bahan baku
Hasil yang diharapkan	Aplikasi <i>mobile</i> dapat menampilkan daftar rekomendasi bahan baku berdasarkan data yang dikirimkan oleh <i>web client</i>

Tabel 6.6 Kasus Uji Menambah Jumlah Bahan Baku

<b>ID Kasus uji</b>	TC-05
<b>Nama kasus uji</b>	Menambah jumlah bahan baku
<b>ID Use case</b>	SRS-04
<b>Tujuan pengujian</b>	Membuktikan bahwa aplikasi <i>mobile</i> dapat menambah jumlah bahan baku berdasarkan hasil perhitungan AHP
<b>Prosedur pengujian</b>	<ol style="list-style-type: none"> <li>1. Berada pada halaman <i>push notification</i></li> <li>2. Menekan salah satu bahan baku</li> <li>3. Memasukkan data ke formulir</li> <li>4. Menekan tombol simpan</li> </ol>
<b>Hasil yang diharapkan</b>	Aplikasi <i>mobile</i> dapat memasukkan data tambahan ke bahan baku

Tabel 6.7 Kasus Uji Mengecek Perhitungan AHP

<b>ID Kasus uji</b>	TC-06
<b>Nama kasus uji</b>	Mengecek perhitungan AHP
<b>ID Use case</b>	SRS-03
<b>Tujuan pengujian</b>	Membuktikan bahwa perhitungan AHP yang ada pada <i>web-client</i> memiliki hasil yang sama dengan perhitungan manualisasi
<b>Prosedur pengujian</b>	Hasil perhitungan AHP pada <i>web-client</i> memiliki hasil yang sama dengan perhitungan manualisasi
<b>Hasil yang diharapkan</b>	Hasil perhitungan AHP pada sistem dan hasil perhitungan AHP pada perhitungan manualisasi memiliki hasil yang sama saat memiliki nilai masukan yang sama

Setelah kasus uji didefinisikan, selanjutnya adalah menentukan hasil pengujian berdasarkan kasus uji pada Tabel 6.1 hingga Tabel 6.7. Hasil pengujian dijabarkan pada Tabel 6.8. Saat hasil yang diharapkan memiliki hasil yang sama dengan hasil yang diharapkan, maka pengujian tersebut bernilai valid. Selain itu, fungsi AHP dan *push notification* diuji dengan keadaan sesungguhnya di kafe, untuk menentukan apakah fungsi AHP dan *push notification* dapat berjalan dengan baik, atau bisa dikatakan valid.

Tabel 6.9 menguji fungsi AHP dan *push notification*. Jika fungsi *push notification* mengirim *push notification* pada saat ada bahan baku habis dan



berisi rekomendasi bahan baku yang harus dibeli, berarti fungsi *push notification* tersebut valid.

**Tabel 6.8 Hasil Pengujian**

ID	Hasil yang diharapkan	Hasil yang diperoleh	Status
TC-01	<i>Web-client</i> dapat memasukkan data pesanan ke <i>database</i>	<i>Web-client</i> dapat memasukkan data pesanan ke <i>database</i>	Valid
TC-02	<i>Web-client</i> dapat memasukkan data bahan baku ke <i>database</i>	<i>Web-client</i> dapat memasukkan data bahan baku ke <i>database</i>	Valid
TC-03	Sistem dapat mengirim <i>push notification</i> ke aplikasi <i>mobile</i> yang berisi hasil rekomendasi perhitungan AHP bila ada bahan baku yang kurang dari minimum	Sistem dapat mengirim <i>push notification</i> ke aplikasi <i>mobile</i> yang berisi hasil rekomendasi perhitungan AHP bila ada bahan baku yang kurang dari minimum	Valid
TC-04	Aplikasi <i>mobile</i> dapat menampilkan daftar rekomendasi bahan baku berdasarkan data yang dikirimkan oleh <i>web client</i>	Aplikasi <i>mobile</i> dapat menampilkan daftar rekomendasi bahan baku berdasarkan data yang dikirimkan oleh <i>web client</i>	Valid
TC-05	Aplikasi <i>mobile</i> dapat memasukkan data tambahan ke bahan baku	Aplikasi <i>mobile</i> dapat memasukkan data tambahan ke bahan baku	Valid
TC-06	Hasil perhitungan AHP pada sistem dan hasil perhitungan AHP pada perhitungan manualisasi memiliki hasil yang sama saat memiliki nilai masukan yang sama	Hasil perhitungan AHP pada sistem dan hasil perhitungan AHP pada perhitungan manualisasi memiliki hasil yang sama saat memiliki nilai masukan yang sama	Valid

Tabel 6.9 memiliki delapan nilai valid dan dua nilai tidak valid. Dua nilai tidak valid tersebut dikarenakan bahan baku yang sama jumlahnya dengan nilai minimum tidak diidentifikasi sebagai bahan yang harus dihitung dengan metode AHP.

Tabel 6.9 Tabel Perbandingan Data Sistem Dengan Data Kafe

No	Nama pemesan	Pesanan	Jumlah	Push notification	Isi push notification	Bahan habis	Valid
1	Kharis	Joes water	1	Tidak ada	-	-	Valid
		Hot choco	1				
2	Tadho	My fair lady	2	Tidak ada	-	-	Valid
		Frappe bubblegum	2				
3	Fikar	Hawaiian sunset	1	Tidak ada	-	-	Valid
		Iced lemon tea	1				
		Frappe bubblegum	2				
4	Bayu	Frappe greentea	2	Tidak ada	-	-	Valid
5	Rani	Frappe bubblegum	2	Tidak ada	-	-	Valid
		My fair lady	2				
6	Sari	Frappe chocolate	3	Tidak ada	-	sirup blueberry lime	Tidak valid
		Joes water	1				
		Hawaiian sunset	2				
7	Aldi			Tidak ada	.	sirup blueberry lime, soda water	Tidak valid
8	Putri	Hot choco	2	Ada	sirup blueberry lime, soda water	sirup blueberry lime, soda water	Valid
9	Nana	Frappe bubblegum	1	Ada	sirup blueberry lime, soda water	sirup blueberry lime, soda water	Valid
		Romantic sparkling	1				
10	Yoan	My fair lady	1	Ada	sirup blueberry lime, soda water	sirup blueberry lime, soda water	Valid

### 6.3 Pengujian akurasi

Pengujian akurasi dilakukan dengan membandingkan data pada sistem dengan pendapat dari pemilik kafe. Tabel 6.10 merupakan perbandingan pendapat antara sistem dengan pendapat pemilik kafe. Terdapat empat pendapat yang berbeda, sehingga akurasi yang didapat adalah 73.33%.

**Tabel 6.10 Perbandingan Pendapat Hari Pertama**

No	Perankingan sistem	Pendapat pemilik	Valid
1	Top	Top	Valid
2	Kitkat	Kitkat	Valid
3	Sponge concentrate	Sponge	Valid
4	Gas whipped	Gas whipped	Valid
5	RAP	Corn flakes	Tidak valid
6	Corn flakes	Diamond ice cream chocolate	Tidak valid
7	Benedico	Benedico	Valid
8	Diamond ice cream chocolate	Diamond ice cream vanilla	Tidak valid
9	Diamond ice cream vanilla	RAP	Tidak Valid
10	Meses	Meses	Valid
11	Puding joy	Puding joy	Valid
12	Syrup hazelnut	Syrup hazelnut	Valid
13	Syrup rum	Syrup rum	Valid
14	Syrup vanilla	Syrup vanilla	Valid
15	Jeruk kaleng	Jeruk kaleng	Valid

**Tabel 6.11 Perbandingan Pendapat Hari Kedua**

No	Perankingan sistem	Pendapat pemilik	Valid
1	Yogurt	Yogurt	Valid
2	Diamond full cream	Diamond full cream	Valid
3	Rice crispy	Koko krunch	Tidak valid
4	Puding jelly happy	Puding jelly happy	Valid
5	Koko krunch	Rice crispy	Tidak valid
6	Oreo	Oreo	Valid
7	Vivo topping whipped cream	Vivo topping whipped cream	Valid
8	Top	top	Valid
9	Kitkat	Kitkat	Valid
10	Mariza strawberry	Sirup mango	Valid
11	Milo professional	Milo professional	Valid
12	Sirup mango	Mariza strawberry	Valid
13	Teh dilmah	Teh dilmah	Valid

Tabel 6.11 merupakan perbandingan pendapat antara sistem dengan pendapat pemilik kafe. Terdapat dua pendapat yang berbeda, sehingga akurasi yang didapat adalah 84.61%. Tabel 6.12 merupakan perbandingan pendapat antara sistem dengan pendapat pemilik kafe. Terdapat tiga pendapat yang berbeda, sehingga akurasi yang didapat adalah 76.92%.

**Tabel 6.12 Perbandingan Pendapat Hari Ketiga**

No	Perankingan sistem	Pendapat pemilik	Valid
1	Jeruk kaleng	Jeruk kaleng	Valid
2	Diamond full cream	Diamond full cream	Valid
3	Astor	Kitkat	Tidak valid
4	Gula	Gula	Valid
5	Puding joy	Puding joy	Valid
6	Sirup strawberry	Sirup strawberry	Valid
7	Tango	Tango	Valid
8	Buah strawberry	Buah strawberry	Valid
9	Kitkat	Powder tiramisu	Tidak valid
10	Mariza coklat	Mariza coklat	Valid
11	Powder tiramisu	Astor	Tidak valid
12	RAP	Rap	Valid
13	Benedico	Benedico	Valid

**Tabel 6.13 Perbandingan Pendapat Hari Keempat**

No	Perankingan sistem	Pendapat pemilik	Valid
1	rice crispy	rice crispy	Valid
2	buah strawberry	buah strawberry	Valid
3	Gula	Gula	Valid
4	Marshmellow	marshmellow	Valid
5	powder bubblegum	powder bubblegum	Valid
6	Oreo	Oreo	Valid
7	vivo topping whipped cream	vivo topping whipped cream	Valid
8	Yogurt	Tango	Tidak valid
9	Astor	Astor	Valid
10	Kitkat	Kitkat	Valid

**Tabel 6.13 Perbandingan Hari Keempat (lanjutan)**

11	Tango	Yogurt	Tidak valid
12	Benedico	mariza caramel	Tidak valid
13	mariza caramel	benedico	Tidak valid

Tabel 6.13 merupakan perbandingan pendapat antara sistem dengan pendapat pemilik kafe. Terdapat empat pendapat yang berbeda, sehingga akurasi yang didapat adalah 69.23%. Tabel 6.14 merupakan perbandingan pendapat antara sistem dengan pendapat pemilik kafe. Terdapat dua pendapat yang berbeda, sehingga akurasi yang didapat adalah 100%.

**Tabel 6.14 Perbandingan Pendapat Hari Kelima**

No	Perankingan sistem	Pendapat pemilik	Valid
1	Kitkat	Yogurt	Valid
2	Tango	Diamond full cream	Valid
3	diamond full cream	Koko krunch	Valid
4	koko crunch	Puding jelly happy	Valid
5	pineapple sunfresh	Rice crispy	Valid

**Tabel 6.15 Perbandingan Pendapat Hari Keenam**

No	Perankingan sistem	Pendapat pemilik	Valid
1	diamond full cream	diamond full cream	Valid
2	buah strawberry	diamond ice cream chocolate	Tidak Valid
3	diamond ice cream chocolate	vivo topping whipped cream	Tidak Valid
4	milo professional	milo professional	Valid
5	vivo topping whipped cream	buah strawberry	Tidak valid
6	corn flakes	corn flakes	Valid
7	koko crunch	koko crunch	Valid
8	Oreo	Oreo	Valid
9	diamond ice cream vanilla	diamond ice cream vanilla	Valid
10	lime sunfresh	lime sunfresh	Valid
11	diamond full cream	diamond full cream	Valid
12	buah strawberry	buah strawberry	Valid
13	diamond ice cream chocolate	diamond ice cream chocolate	Valid

Tabel 6.15 merupakan perbandingan pendapat antara sistem dengan pendapat pemilik kafe. Terdapat tiga pendapat yang berbeda, sehingga akurasi yang didapat adalah 76.92%. Tabel 6.16 merupakan perbandingan pendapat antara sistem dengan pendapat pemilik kafe. Terdapat dua pendapat yang berbeda, sehingga akurasi yang didapat adalah 75%.

**Tabel 6.16 Perbandingan Pendapat Hari Ketujuh**

No	Perankingan sistem	Pendapat pemilik	Valid
1	diamond full cream	diamond full cream	Valid
2	buah strawberry	buah strawberry	Valid
3	vivo topping whipped cream	Tango	Tidak valid
4	Oreo	Oreo	Valid
5	Tango	vivo topping whipped cream	Tidak valid
6	jeruk kaleng	jeruk kaleng	Valid
7	diamond ice cream vanilla	diamond ice cream vanilla	Valid
8	the dilmah	the dilmah	Valid

Pada Tabel 6.17, nilai persentase yang didapatkan hanya 68.44% karena basis pendapat pemilik kafe memiliki perberbedaan dengan kriteria yang ada pada sistem. Perbedaan basis pertimbangan terletak pada pemilik kafe yang juga mempertimbangkan bahan yang digunakan pada menu yang sedang populer pada saat itu, yang selalu berubah terus-menerus pada waktu yang tidak tentu, sehingga nilai akurasi rata-rata selama tujuh hari adalah 68.44%.

**Tabel 6.17 Rata-rata akurasi**

Hari	Akurasi
1	73.33%
2	84.61%
3	76.92%
4	69.23%
5	100%
6	76.92%
7	75%
Rata-rata	68.44%



## BAB 7 PENUTUP

Bab penutup berisikan kesimpulan dari hasil penelitian dan saran. Kesimpulan menjawab pertanyaan yang ada pada rumusan masalah.

### 7.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil analisis kebutuhan, perancangan, implementasi dan pengujian pada rancang bangun aplikasi pengelolaan bahan baku kafe adalah:

1. Metode AHP diimplementasikan dalam sistem pengelolaan bahan baku kafe untuk merekomendasikan bahan baku yang harus dibeli. Melalui AHP, dilakukan pembobotan terhadap tiga kriteria yang ada, yaitu sisa bahan, jumlah menu yang digunakan oleh bahan tersebut, dan harga. Hasil bobot digunakan untuk membuat prioritas berupa bahan baku yang ada.
2. Aplikasi pengelolaan bahan baku kafe diimplementasikan pada perangkat bergerak *smartphone* dengan sistem operasi Android. Bahasa yang digunakan adalah Java. Untuk aplikasi *web-client* yang digunakan oleh administrator, menggunakan Bahasa pemrograman HTML, CSS, dan Javascript. Aplikasi *web-client* dapat mengirim *push notification* melalui Firebase Cloud Messaging, menggunakan Firebase Cloud Functions sebagai trigger. Dari hasil pengujian validasi *push notification* pada aplikasi pengelolaan bahan baku kafe menggunakan *blackbox*, diperoleh hasil persentase 80%. Persentase pengujian validasi *push notification* hanya 80% karena terdapat bahan baku yang nilai sisanya sama dengan nilai minimum dan tidak terdeteksi oleh sistem.
3. Dari hasil pengujian AHP yang membandingkan kerja sistem dengan pendapat pemilik kafe, didapatkan hasil persentase 68.44%. Persentase yang didapat tidak terlalu tinggi karena terdapat perbedaan kriteria sistem dengan pertimbangan pribadi pemilik kafe.

### 7.2 Saran

Berikut saran yang diberikan untuk pengembangan aplikasi pengelolaan bahan baku kafe menggunakan AHP dan *push notification* antara lain:

1. Dapat ditambahkan fitur untuk mengelola menu, sehingga saat pemilik kafe ingin mengubah menu kafanya dapat dilakukan dengan cepat.
2. Aplikasi dapat dikembangkan dengan memberikan fitur kasir dengan dasar fitur pemesanan sehingga lebih mudah dalam penggunaannya.
3. Dapat ditambahkan metode-metode yang lain agar dapat menaikkan tingkat akurasi perhitungan AHP pada sistem.

## DAFTAR PUSTAKA

- Ardiyanto, H. (2013). Sistem Pendukung Keputusan Pemilihan Perumahan menggunakan Metode AHP berbasis Web (Studi Kasus CV. Wlsma Anungkriya Demak). *Journal of Informatics and Technology*, 2, 50-58.
- Ariff, H., Salit, M., Ismail, N., & Nukman, Y. (2008). Use of Analytical Hierarchy Process (AHP) for Selecting the Best Design Concept. *Jurnal Teknologi*, 49, 1-18.
- Bagus Prasetyo, W. L. (2013). Sistem Pendukung Keputusan Pemilihan Paket INternet Operator Telekomunikasi dengan Metode AHP (analytical Hierarchy Process). *Jurnal TIKomSin*, 2338-4018, 7-12.
- Developers, G. (2018). *Firebase*. Retrieved Juni 2, 2018, from <https://firebase.google.com/docs/database/ios/structure-data>
- Developers, G. (2018). *Firebase*. Retrieved Juni 5, 2018, from <https://firebase.google.com/docs/cloud-messaging/>
- Google Cloud Messaging for Android. (n.d.). Retrieved Januari 24, 2015, from <http://developer.android.com/google/gcm/index.html>
- L.Saaty, T. (2008). Decision Making With The Analytical Hierarchy Process. *Int. J. Services Sciences*, 1, 83-98.
- Maulidi, A. (2017). *Pengertian Kafe (Cafe)*. Retrieved Agustus 9, 2017, from <http://www.kanalinfo.web.id/2017/02/pengertian-kafe-cafe.html>
- Nations, D. (n.d.). *What is Push Notification*. Retrieved Oktober 2014, 2014, from <http://ipad.about.com/od/iPad-Glossary/g/What-Is-Push-Notification.htm>
- Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta, Dr. Asoke Nath. (2018). Real-Time Communication Application Based on Android using Google Firebase. *International Journal of Advance Research in Computer Science and Management Studies*, 6(4), 75-79.
- Norddin, N., Ibrahim, K., & Aziz, A. (2012). Selecting New Lecturers Using The Analytical Hierarchy Process. *Statistic in Science, Business, and Engineering (ICSSBE)*.
- Oktariado, O. (2009). Penentuan Peringkat Bahaya Tsunami dengan Metode Analytical Hierarchy Process (Studi Kasus: Wilayah Pesisir Kabupaten Sukabumi). *Jurnal Geologi Indonesia*, 4(no.2), 103-116.
- Random Log. (2010). *Pengertian Push (Email/Notification)*. Retrieved Januari 24, 2015, from <http://www.randomlog.org/article/pengertian-push-emailnotification/>

- Saaty, T. L. (1999). Basic Theory Of The Analytic Hierarchy Process: How To Make A Decision. *Rev.R.Acad.Cienc.Exact.Fis.Nat. (Esp)*, 93, 395-423.
- Sonam Khedkar, Swapnil Thube. (2017). Real Time Databases for Applications. *International Research Journal of Engineering and Technology (IRJET)*, 4(6), 2078-2082.
- Tanjung, F. A. (2013). *Tren Bisnis Kafe, Terutama di Indonesia*. Retrieved January 25, 2015, from <http://www.bglconline.com/2013/02/tren-bisnis-kafe-terutama-di-indonesia/>
- Tolle, H. (2014). *Notifikasi Push Android menggunakan Google Cloud Messaging (GCM), PHP dan MySQL*. Retrieved Februari 10, 2015, from <http://www.hermantolle.com/class/2014/11/notifikasi-push-android-menggunakan-google-cloud-messaging-gcm-php-dan-mysql/>

